

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-095874

(43)Date of publication of application : 08.04.1994

(51)Int.CI. G06F 9/30
G06F 9/34

(21)Application number : 05-165052 (71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 11.06.1993 (72)Inventor : JAMES EDWARD PHILIP VASSILIADIS STAMATIS

(30)Priority

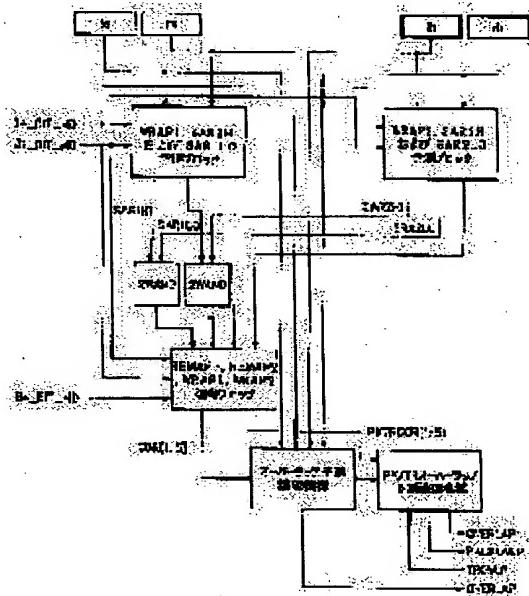
Priority number : 92 920941 Priority date : 28.07.1992 Priority country : US

(54) DIGITAL COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To provide a device which detects the stored operand overlap for SS instruction which has the same overlap detection condition as a character movement (MVC) instruction and to prove that this device is effective for this detection.

CONSTITUTION: This device is applied to all ESA/390 addressing modes including an access register addressing mode for 24-bit or 31-bit addressing. 24-Bit or 31-bit S/370 addressing is handled as a special case of access register addressing. Further, this device is extended so as to support other addressing modes including the 64-bit addressing mode. This device is provided with a high-speed parallel execution mode also.



LEGAL STATUS

[Date of request for examination] 11.06.1993

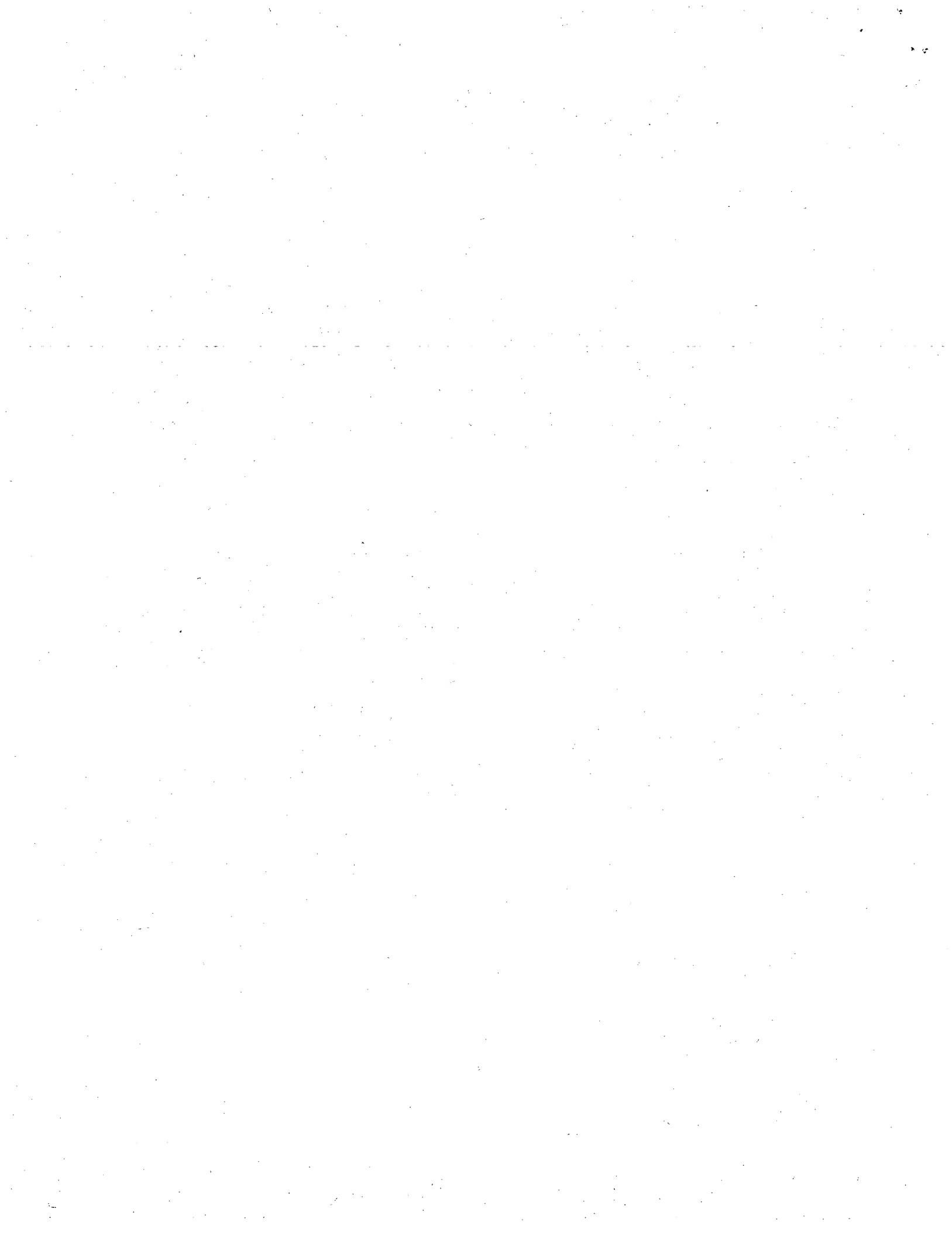
[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2500098

[Date of registration] 01.03.1996



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-95874

(43)公開日 平成6年(1994)4月8日

(51)Int.Cl. ⁵ G 0 6 F 9/30 9/34	識別記号 3 5 0 B 9189-5B 3 2 0 A 9189-5B	序内整理番号 F I	技術表示箇所
--	--	---------------	--------

審査請求 有 請求項の数12(全42頁)

(21)出願番号 特願平5-165052
(22)出願日 平成5年(1993)6月11日
(31)優先権主張番号 920941
(32)優先日 1992年7月28日
(33)優先権主張国 米国(US)

(71)出願人 390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MACHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州
アーモンク(番地なし)
(72)発明者 ジェームズ・エドワード・フィリップス
アメリカ合衆国13905、ニューヨーク州ビンガムトン、クレイリー・アベニュー138
(74)代理人 弁理士 合田 淩(外1名)

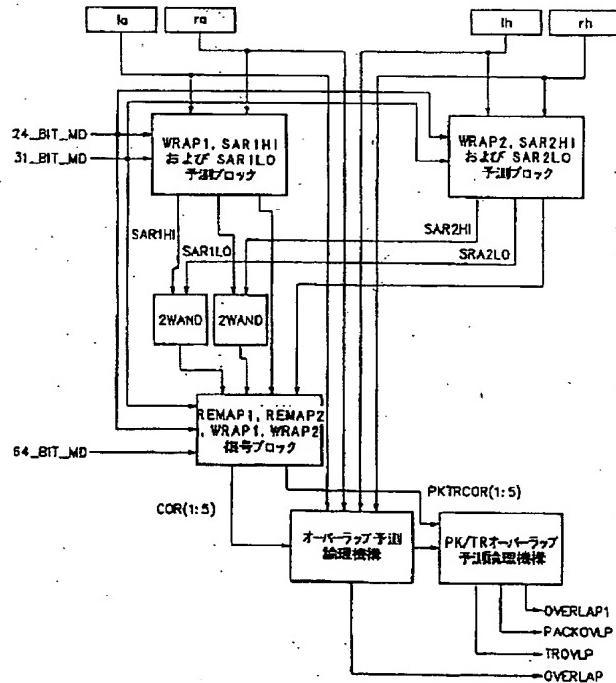
最終頁に続く

(54)【発明の名称】 デジタル・コンピュータ・システム

(57)【要約】 (修正有)

【目的】 文字移動(MVC)命令と同じオーバーラップ検出条件を有するSS命令用の記憶オペランド・オーバーラップを検出するための装置を提示し、それに有効なことを証明すること。

【構成】 この装置は、24ビットまたは31ビット・アドレス指定用のアクセス・レジスタ・アドレス指定を含むすべてのESA/390アドレス指定モードに適用できる。24ビット及び31ビットのS/370アドレス指定もアクセス・レジスタ・アドレス装置の特殊なケースとして扱われる。さらに、この装置は、他のアドレス装置モードをサポートするように拡張され、64ビット・アドレス指定モードを含む例が示される。この装置の高速並列実施態様も提示される。



【特許請求の範囲】

【請求項1】記憶オペランド・オーバーラップ予測機構と、

宛先記憶オペランド・アドレス低及び折返し検出器と、ソース記憶オペランド・アドレス高及び折返し検出器と、

オーバーラップ検出器用の制御論理機構とを備えるデジタル・コンピュータ・システム。

【請求項2】オーバーラップが、宛先記憶オペランド・アドレスとソース記憶オペランド・アドレスの生成に使用される入力ならびにオペランド長さから予測され、ソース記憶オペランド・アドレス及び宛先記憶オペランド・アドレスの計算と並列に実行されることを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項3】前記宛先記憶オペランド・アドレス低及び折返し検出器中で、宛先オペランド・アドレスが仮想ア

$$SAR1H_i = Ia, \forall r a_{(i-52)}$$

$52 \leq i \leq 55$

$$SAR1G_i_N = \overline{Ia_i \ ra_{(i-52)}}$$

$52 \leq i \leq 62$

$$SAR1T_i_N = \overline{Ia_i + ra_{(i-52)}}$$

$52 \leq i \leq 62$

$$SAR1C_{63}_N1 = \overline{Ia_{63} \ ra_{51}}$$

$52 \leq i \leq 55$

$$SAR1H_i_N = \overline{SAR1H_i}$$

$$(SAR1G)_i^{+1} = (SAR1G)_i_N ((SAR1T)_i_N + (SAR1G)_{i+1}_N)$$

$52 \leq i \leq 61$

$$(SAR1T)_i^{+1} = ((SAR1T)_i_N + (SAR1T)_{i+1}_N)$$

$52 \leq i \leq 61$

$$(SAR1C1)_{62} = (SAR1G)_{62}_N ((SAR1T)_{62}_N + (SAR1C)_{63}_N1)$$

$$(SAR1C1)_{63} = \overline{(SAR1C)_{63}_N1}$$

$$(SAR1G)_i^{-1}_N = \overline{(SAR1G)_i^{+1} + (SAR1T)_i^{+1} (SAR1G)_{i+1}^{+1}}$$

$52 \leq i \leq 59$

$$(SAR1T)_i^{+3}_N = \overline{(SAR1T)_i^{+1} (SAR1T)_{i+2}^{+1}}$$

$52 \leq i \leq 59$

$$(SAR1C)_i_{N2} = \overline{(SAR1G)_i^{+1} + (SAR1T)_i^{+1} (SAR1C1)_{i+2}}$$

$i = 60, 61$

$$(SAR1C)_i_{N2} = \overline{(SAR1C1)_i}$$

$i = 62, 63$

$$(SAR1G)_i^{+3}_N = \overline{(SAR1G)_i^{+1} ((SAR1T)_i^{+3}_N + (SAR1G)_{i+1}^{+1}_N)}$$

$52 \leq i \leq 55$

$$(SAR1T)_i^{+3} = \overline{(SAR1T)_i^{+3}_N + (SAR1T)_{i+2}^{+1}_N}$$

$52 \leq i \leq 55$

$$(SAR1C)_{56} = \overline{(SAR1G)_{56}_N ((SAR1T)_{56}_N + (SAR1C)_{50}_N2)}$$

56

$$(SAR1C)_i_{N2} = \overline{(SAR1C)_i_{N2}}$$

$60 \leq i \leq 63$

$$(SAR1C)_i_{N} = \overline{(SAR1G)_i^{-1} + (SAR1T)_i^{+1} (SAR1C)_{i+1}}$$

$52 \leq i \leq 55$

$$(SAR1C)_{56}_N = \overline{(SAR1C)_{56}}$$

$52 \leq i \leq 55$

$$SAR1_i_N = (SAR1C)_{i+1}_N \vee SAR1H_i$$

$52 \leq i \leq 55$

$$SAR1_i = (SAR1C)_{i+1}_N \vee SAR1H_i_{N}$$

$52 \leq i \leq 55$

$$SAR1_{52_to_55}_ZEROS = (SAR1_{52}_N) (SAR1_{53}_N) (SAR1_{54}_N) (SAR1_{55}_N)$$

1は緩衝キャリーなしを表し、(SAR1C1)_iは緩

$$SAR1_{52_to_55}_ONES = (SAR1_{52}) (SAR1_{53}) (SAR1_{54}) (SAR1_{55})$$

衝キャリーを表し、(SAR1C)_iはキャリーを表

$$(SAR1C)_{52} = \overline{(SAR1C)_{52}_N}$$

し、(SAR1C)_i_Nはキャリーなしを表し、SA

上式において、SAR1H_iは、ビット位置_iにおける半和を表し、SAR1G_i_Nは生成なしを表し、SAR1Ti_Nは伝送なしを表し、(SAR1C)_i_N

R1_TO_51_OS は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力1_a がすべて0であることを表し、SAR1_TO_51_1S は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力1_a がすべて1であることを表し、SAR1_TO_50_1S_A_51_0 は、24ビット・アドレス指定では40と50の間の両端を含むすべてのビット位置で入力1_a がすべて1でビット位置51では1であり、31ビット・アドレス指定では33と50の間の両端を含むすべてのビット位置で入力1_a がすべて1でビット位置51では1であることを表し、SAR1LO は、SAR1 が仮想アドレス空間の低位256バイト・ブロック中にあることを表し、WRAP1 は、ソース・アドレス計算がアドレス計算中に高位仮想メモリから低位仮想メモリへ折り返すことを表すことを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

* 10

*【請求項7】前記ソース記憶オペランド・アドレス高及び折返し検出器中で、ソース・オペランド・アドレスが仮想アドレスの高位256バイト・ブロック中にあること、及び仮想アドレス空間の高位4Kブロックから仮想アドレス空間の低位4Kブロックへの折返し動作がアドレス生成中に起こるかどうかが判定されることを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項8】前記ソース記憶オペランド・アドレス高及び折返し検出器中で、64ビット入力1_h に宛先記憶オペランド・アドレスの算出に使用される基底が供給され、12ビット入力r_h に同じく変位が供給されることを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項9】前記ソース記憶オペランド・アドレス高及び折返し検出器が12ビット加算器を含み、その中の挙動が下記の式によって決定され、

【数7】

$$\begin{aligned}
 SAR2H_i &= \overline{lh_i \vee rh_{(i-52)}} & 52 \leq i \leq 55 \\
 SAR2G_i_N &= \overline{lh_i \vee rh_{(i-52)}} & 52 \leq i \leq 62 \\
 SAR2T_i_N &= \overline{lh_i + rh_{(i-52)}} & 52 \leq i \leq 62 \\
 SAR2C_{62}_N1 &= \overline{lh_{62} \vee rh_{11}} & \\
 SAR2H_i_N &= \overline{\overline{SAR2H}_i} & 52 \leq i \leq 55 \\
 (SAR2G)_{i+1} &= \overline{(SAR2G)_i_N ((SAR2T)_i_N + (SAR2G)_{i+1}_N)} & 52 \leq i \leq 61 \\
 (SAR2T)_{i+1} &= \overline{((SAR2T)_i_N + (SAR2T)_{i+1}_N)} & 52 \leq i \leq 61 \\
 (SAR2C1)_{62} &= \overline{(SAR2G)_{62}_N ((SAR2T)_{62}_N + (SAR2C)_{63}_N1)} & \\
 (SAR2C1)_{63} &= \overline{(SAR2C)_{63}_N1} & \\
 (SAR2G)_{i+3}_N &= \overline{(SAR2G)_{i+1}_N + (SAR2T)_{i+1}_N + (SAR2G)_{i+2}_N} & 52 \leq i \leq 59 \\
 (SAR2T)_{i+3}_N &= \overline{(SAR2T)_{i+1}_N + (SAR2T)_{i+2}_N} & 52 \leq i \leq 59 \\
 (SAR2C)_{i+2}_N2 &= \overline{(SAR2G)_{i+1}_N + (SAR2T)_{i+1}_N + (SAR2C1)_{i+2}_N} & i = 60, 61 \\
 (SAR2C)_{i+2}_N1 &= \overline{(SAR2C1)_{i+2}_N} & i = 62, 63 \\
 (SAR2G)_{i+3}_N &= \overline{(SAR2G)_{i+3}_N ((SAR2T)_{i+3}_N + (SAR2G)_{i+4}_N)} & 52 \leq i \leq 55 \\
 (SAR2T)_{i+3}_N &= \overline{(SAR2T)_{i+3}_N + (SAR2T)_{i+4}_N} & 52 \leq i \leq 55 \\
 (SAR2C)_{56} &= \overline{(SAR2G)_{56}_N ((SAR2T)_{56}_N + (SAR2C)_{56}_N2)} & 56 \\
 (SAR2C)_{56} &= \overline{(SAR2C)_{56}_N2} & 60 \leq i \leq 63 \\
 (SAR2C)_{i+1}_N &= \overline{(SAR2G)_{i+3}_N + (SAR2T)_{i+3}_N + (SAR2C)_{i+4}_N} & 52 \leq i \leq 55 \\
 (SAR2C)_{56}_N &= \overline{(SAR2C)_{56}_N2} & \\
 SAR2_i_N &= \overline{(SAR2C)_{i+1}_N \vee SAR2H_i} & 52 \leq i \leq 55 \\
 SAR2_i &= \overline{(SAR2C)_{i+1}_N \vee SAR2H_i_N} & 52 \leq i \leq 55 \\
 SAR2_52_to_55_ZEROS &= (SAR2_{52}_N) (SAR2_{53}_N) (SAR2_{54}_N) (SAR2_{55}_N) \\
 SAR2_52_to_55_ONES &= (SAR2_{52}) (SAR2_{53}) (SAR2_{54}) (SAR2_{55}) \\
 (SAR2C)_{52} &= \overline{(SAR2C)_{52}_N}
 \end{aligned}$$

上式において、SAR2H_i は、ビット位置iにおける半和を表し、SAR2G_i_N は生成なしを表し、SAR2T_i_N は生成ありを表す。

9

ット位置で入力l h がすべて1であることを表し、24_BIT_MDは、24ビット・アドレス指定モードが有効であることを表し、31_BIT_MDは31ビット・アドレス指定モードが有効であることを表し、SAR2_TO_51_OSは、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力l h がすべて0であることを表し、SAR2_TO_51_1Sは、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力l h がすべて1であることを表し、SAR2_TO_50_1S_A_51_0は、24ビット・アドレス指定では40と*

$$\text{REMAP1_N} = \overline{\text{SAR2HI}} \text{ } \overline{\text{SAR1LO}}$$

$$\text{REMAP1} = \overline{\text{REMAP1_N}}$$

$$\text{COR(1)} = 64_BIT_MD + \text{REMAP1_N WRAP1_N WRAP2_N} + \text{WRAP2 WRAP1}$$

$$\text{COR(2)} = 24_BIT_MD \text{REMAP1_N WRAP2_N WRAP1}$$

$$\text{COR(3)} = 24_BIT_MD \text{WRAP2 WRAP1_N} + 24_BIT_MD \text{REMAP1 WRAP1_N}$$

$$\text{COR(4)} = 31_BIT_MD \text{REMAP1_N WRAP2_N WRAP1}$$

$$\text{COR(5)} = 31_BIT_MD \text{WRAP2 WRAP1_N} + 31_BIT_MD \text{REMAP1 WRAP1_N}$$

上式において、24_BIT_MDは、24ビット・アドレス指定が有効なことを表し、31_BIT_MDは、31ビット・アドレス指定が有効なことを表し、REMAP1とREMAP1_Nは、宛先記憶アドレスが仮想アドレス空間の低位256バイト・ブロックにあり、ソース記憶アドレスが仮想アドレス空間の高位256バイト・ブロックにある真と補を表し、COR(1)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第1の計算を表し、COR(2)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第2の計算を表し、COR(3)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第3の計算を表し、COR(4)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第4の計算を表し、COR(5)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第5の計算を表すことを特徴とする、請求項1に記載のディジタル・コンピュータ・システム。

【請求項12】前記記憶オペランド・オーバーラップ予測機構が、

入力l a、l h、ra、rh、COR(1:5)を備える予測論理機構と、

複数のキャリーセーブ加算器CSA1、CSA2、CSA3、CSA4、CSA5、CSA6、CSA7と、キャリー先読み加算器CLA1と、

キャリーアンダーフロー生成器CG1と、

複数の排他的論理和ブロックXR7、XR8と、

複数の多元ANDブロックA1、A2、A3、A4、A 50

10

*50の間の両端を含むすべてのビット位置で入力l h がすべて1でありビット位置51では1であり、31ビット・アドレス指定では33と50の間の両端を含むすべてのビット位置で入力l h がすべて1でありビット位置51では1であることを表し、SAR2_HIは、SAR2が仮想アドレス空間の低位256バイト・ブロック中にあることを表し、WRAP2は、ソース・アドレス計算がアドレス計算中に高位仮想メモリから低位仮想メモリへ折り返すことを特徴とする、請求項1に記載のディジタル・コンピュータ・システム。

【請求項11】前記オーバーラップ検出器用の制御論理の挙動が下記の式式1-3のブール式によって決定され、

【数式1-3】

5と、

OVERLAPを計算するための多重化論理機構とを有することを特徴とする、請求項1に記載のディジタル・コンピュータ・システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ディジタル・コンピュータ・システムに関し、詳細にはソース位置と宛先位置の破壊的オーバーラップを予測し、それによって条件の決定が計算できるようになるまでパイプライン式ディジタル・コンピュータ・システムにおいて実行を停止する必要なく、即時に移動が実行できるようにすることにより、ある記憶位置から第2の記憶位置への頻繁に使用される移動の実行を改善することに関する。

【0002】

【従来の技術】はじめに、本明細書で用いる用語について説明する。本明細書で用いる一部の用語には辞書上の意味も含まれるが、以下のいくつかの用語の用語集が役立つであろう。

【0003】ESA/390は、現在使用されているエンタープライズ・システム体系で使用される、インターナショナル・ビジネス・マシンズ・コーポレーションのメインフレーム・システム・アーキテクチャを指す。このアーキテクチャは、世界中のメインフレーム・アーキテクチャの標準となっており、既存の命令を使用することが、既存のアーキテクチャの使用を可能にする上で重要である。

【0004】MVCは、既存のESA/390命令の1

11

つであるMOVE CHARACTER(文字移動)の略号である。

【0005】NCは、既存のESA/390命令の1つであるAND CHARACTER(文字論理積)の略号である。

【0006】OCは、既存のESA/390命令の1つであるOR CHARACTER(文字論理和)の略号である。

【0007】XORは、既存のESA/390命令の1つであるXOR CHARACTER(文字排他的論理和)の略号である。
10

【0008】MVZは、既存のESA/390命令の1つであるMOVE ZONES(ゾーン移動)の略号である。

【0009】MVNは、既存のESA/390命令の1つであるMOVE NUMERICS(数値移動)の略号である。

【0010】SAR1は、第1記憶オペランド・アドレスの略号である。

【0011】SAR2は、第2記憶オペランド・アドレスの略号である。
20

【0012】AGENとAGは、アドレス生成の略語として同じ意味で使用する。

【0013】次に、従来の技術について述べる。本発明の背景として、読者がこの議論を速やかに理解し、本発明が従来の技術とどう異なるか理解できるように、IBMテクニカル・ディスクロージャ・ブルテン所載のいくつかの論文を簡単に紹介する。まず、本発明は、MVC命令、ならびにMVC命令と同様の形で潜在的にオーバーラップする可能性のあるオペランドを有することできる命令について、破壊的な記憶オペランドのオーバーラップを検出するための予測機構を提供するものであることを理解されたい。本発明は、ESA/390プロセッサ・アキテクチャで使用されるタイプのアドレス計算における破壊的なオペランドのオーバーラップの予測に適用される。本発明が、他のアキテクチャのアドレス指定モードをサポートできるように容易に拡張できることは、想定した64ビットのアドレス指定モードを含むように検出を拡大することによって実証される。
30

【0014】インターナショナル・ビジネス・マシンズ・コーポレーション(以下ではIBMと称する)は、IBM社内で行った発明を一般に知らせるため、テクニカル・ディスクロージャ・ブルテン(TDB)と称する雑誌を発行している。TDBで当技術分野に関する論文を探してみると、TDB12-82でガイ(Ngai)等が、破壊的オペランド・オーバーラップ移動命令の実行を扱うための高速アルゴリズムを発表していることに気付く。このアルゴリズムは、MOVE CHARACTER、MOVE NUMERICS、MOVE ZONESなどのタイプの命令に適している。しかし、破壊
40

12

的オペランド・オーバーラップ条件の決定は仮定されており、明記されていない。この仮定条件の予測が本発明の主題である。

【0015】ビーチャー(Beetcher)等はTDB06-81で、命令の挙動が、オペランドがオーバーラップしているかどうかに依存する、Move Long Embellished(MLE)命令を定義している。この命令定義はこのTDB論文に明記されているが、記憶オペランドのオーバーラップを検出するための機構は定義されていない。

【0016】ガイ等はTDB06-76で、宛先アドレスとソース・アドレスのビットを使ってROS(書き込み専用記憶装置)にアクセスすることにより、破壊的な記憶オペランドのオーバーラップを判定するハードウェア機構を発表している。このROSは、記憶オペランドのオーバーラップがその組合せから発生する可能性があるかどうかを示す標識を各記憶位置に格納している(過剰指示)。アドレスの全ビットが使用される。複数のアドレス指定モードのサポートについての説明はない。また、ガイ等の方法では、計算機のデータフロー幅に関する条件が存在するかどうかを判定する。本発明では、アキテクチャによって示される所によれば記憶オペランドが破壊的なオペランドのオーバーラップを有するかどうかを厳密に予測する。この条件は、オペランド内の任意のバイトがオーバーラップする場合に発生する。トレース解析によれば、現在の高性能プロセッサ実施態様で普通と見られるダブルワードの記憶データ・バス幅よりもオペランド長が短い場合、オペランドがオーバーラップする条件はまれである。したがって、一般的なケースについてオーバーラップを検出することによって、性能は余り低下しない。いずれにせよ、本発明の指示は厳密であり、過剰指示ではない。第2に、本発明では、その条件を完全に論理によって検出し、アレイ・アクセスには頼らない。したがって、本発明の決定は、ガイ等が発表した機構よりも高速でセル効率が高くなるはずである。最後に、ガイ等が発表した機構では、本発明のように基底、変位およびオペランド長からではなく、生成された後のアドレスからオーバーラップ条件を予測している。

【0017】ブルックス(Brooks)はTDB06-76で、データ・フィールドに対する両立できない同時アドレスを防止するためのオペランド整合性アテンダント(OCA)を発表している。この論文で記述されるハードウェア機構では、開始オペランド・アドレスと終了オペランド・アドレスの比較によって、オペランドがオーバーラップするかどうかを判定する。したがって、決定の前にオペランドの開始アドレスだけでなく終了アドレスも計算しておく必要がある。この開始アドレスと終了アドレスが得られた後に、オペランドのオーバーラップが存在することを近似して、比較の速度を上げる機構が利用できる。一方、本発明では、ソース・アドレス及び宛
50

13

先アドレス(上記TDB論文の開始アドレス)の生成と並行して、記憶オペランド・オーバーラップ条件を判定する。さらに、本発明の指示は厳密である。

【0018】ベーカー(Baker)等はTDB 01-82で、オペランドがオーバーラップしたとき、オペランドがバイトごとに処理されるかのようにその挙動が進行する、多くの命令を実行するためのアルゴリズムを発表した。オペランド・オーバーラップの検出が既知であり、実行のためのアルゴリズムが準備されていると仮定されている。本発明の主題である、オペランド・オーバーラップ条件を検出するための機構は示されていない。

【0019】

【発明が解決しようとする課題】本発明は、2つの記憶オペランドが破壊的にオーバーラップするかどうかを予測するために、記憶アドレスの生成と並行して実行する、特殊なハードウェアの使用に関する。記憶装置間命令の実行の挙動は、その命令が記憶オペランドに対してバイトごとに実行されるかのように進むようなアーキテクチャになっていることがしばしばである。これは、これらすべての動作をバイトごとに実行することによって達成できる。しかし、ソース・オペランドと宛先オペランドが破壊的にオーバーラップしない場合は、複数のバイトに対して並列に動作を実行し、なおかつそれがバイトごとの実行のように見えるようにすることができる。したがって、記憶オペランドが破壊的にオーバーラップすることを検出し、そうでない場合は一般に複数のバイトに対して実行を進めさせ、オペランドが破壊的にオーバーラップする場合は動作をバイトごとに実行することにより、命令の実行速度を高めることができる。アーキテクチャ上の要件を満たすには、取り出したデータが記憶される前に、破壊的オペランド・オーバーラップ条件がわかっていないなければならない。場合によっては、記憶オペランドのオーバーラップが判定されている間に有用な作業を実施することができるが、頻繁に使用されるMOVE CHARACTER(MVC)命令では、このギャップを埋める有用な作業はない。したがって、オペランド・オーバーラップ条件が判定されるまで、実行を停止しなければならない。

【0020】

【課題を解決するための手段】従来、破壊的記憶オペランド・オーバーラップを検出するには、まずソース・オペランドと宛先オペランドの記憶アドレスを計算し、統いて両記憶アドレスを比較して、次式が成立するかどうかを判定していた。

【数14】 $0 \leq S\text{AR}_1 - S\text{AR}_2 \leq L$

【0021】上式で、SAR1とSAR2は、それぞれ宛先オペランドとソース・オペランドの記憶オペランド・アドレスを表し、Lはオペランドの長さを表す。バイナリ式プロセッサでは、この検出に、記憶アドレスの生成に加えて少なくとももう1サイクルを要した。し

10

20

30

40

50

14

たがって、MVC命令が進行できるようになる前に少なくとも1停止サイクルが必要である。本発明によると、MVC命令のソース・オペランドと宛先オペランドの基底と変位、および命令中で指定されるオペランドの長さから直接にMVC命令の記憶オペランド・オーバーラップ条件を判定することができる。この判断をソース記憶アドレス及び宛先記憶アドレスの生成と並行して実行することにより、頻繁に呼び出されるMVC命令の実行中に停止サイクルが必要でなくなる。MVC命令について1停止サイクル節約されるだけで、オペランド長が8バイト未満の比較的よく見られるケース(84%)ではMVC命令の実行速度が21%向上し、オペランド長が8バイトを超えるケース(16%)では9%向上する。MVC命令の頻度により、上記の数字は、代表的命令混合体でCPIの純減として反映され、MVC命令だけの場合はそれが0.02CPIとなり、しかも最小のハードウェアを追加するだけで達成される。本発明は、ESA/390プロセッサ用に設計されており、そのアーキテクチャにとって既知のすべてのアドレス指定モードについて破壊的記憶オペランドを予測することができる。また想定される64ビットのアドレス指定モードをサポートするように予測機構を拡張することにより、本発明を他のアーキテクチャに拡張できることも実証されている。

【0022】上記その他改善点は、下記の詳しい説明に示してある。本発明とその利点および特徴をよりよく理解するために、以下の説明および図面を理解されたい。

【0023】

【実施例】本発明の好ましい実施例を詳細に考察する前に、ソース・アドレスと宛先アドレスの破壊的オーバーラップを予測するのに使用される条件を展開しておくのが有意義であろう。

【0024】IBM ESA/390などのアーキテクチャでは、オーバーラップする記憶オペランドに対する命令の実行をアーキテクチャ方式で実施することが必要な場合がある。それには、命令の実行を開始する前にそのようなオーバーラップ条件の検出が必要となることがあり得る。これらのタイプの命令のいくつか(主な例はESA/390におけるMOVE CHARACTER(MVC)命令)の頻度と、その命令の大部分の呼出しの際にその実行に必要なサイクル数とが与えられているものとすると、このプロセスをオペランド・オーバーラップの順次検出によって実施する際には、大きな不利益を受ける恐れがある。このことは、オペランド・オーバーラップの速やかな判定が必要なことを示唆している。前記のS命令MVC、およびESA/390アーキテクチャで見られる他のSS命令AND CHARACTER(NC)、OR CHARACTER(OC)、XOR CHARACTER(XC)、MOVE ZONES(MVZ)、MOVE NUMERIC(MVN)

15

では、オペランドのオーバーラップする条件は次式で表すことができる。

【数15】 $0 \leq SAR_1 - SAR_2 \leq L$

【0025】上式で、 SAR_1 と SAR_2 は、それぞれ宛先オペランドとソース・オペランドの記憶アドレスを表し、 L はオペランドの長さを表す。上記のリスト中の命令のうちでMVCが命令ストリームの中で最も頻繁に発生する。したがって、上記のオーバーラップを、MVC型オーバーラップと呼ぶことにする。さらに、他のES *

$SAR_2 \leq SAR_1 + L_1 \leq SAR_2 + (2 * L_2) - 2$

【数18】

$SAR_2 \leq SAR_1 + L_1 \leq SAR_2 + 255$

【0026】MVC型オーバーラップは厳密に検出しなければならないが、後者の命令群でのオーバーラップは過剰指示することができる。オーバーラップの過剰指示は、PACK、ZAPおよびUNPKでは数式19によって実施でき、TRとTRTでは数式20によって実施できる。

【数19】 $|SAR_1 - SAR_2| \leq 15$

【数20】 $|SAR_1 - SAR_2| \leq 255$

【0027】前述のように、MVCは、代表的な命令混合体で最も頻繁に現れる命令の1つである。したがって、設計が与えられている場合、オーバーラップの検出に使用される機構が、命令当たりサイクル数(CPI)に影響を与えることがあり得る。従来技術では、MVCオーバーラップ検出は、両方の記憶オペランド・アドレス SAR_1 と SAR_2 を計算し、続いて SAR_1 を SAR_2 と、また $SAR_1 - SAR_2$ を L と比較することからなっていた。このような実施態様は記憶バスの過小利用をもたらし、性能上のボトルネックを生じる可能性がある。というのは、 SAR_1 と SAR_2 の計算および SAR_1 と SAR_2 の比較にはAGEN ALUが必要だからである。オーバーラップの検出から生じ得る性能低下の例として、単一サイクル命令の場合は

ID-AG-EX

記憶アクセスを含む2サイクル命令の場合は

ID-AG-CA

μF -AG-EX

のパイプライン構造を使用する、高性能スーパースカラー・プロセッサ用のPU設計を考えてみる。上式で μF はマイクロワード取出しを表す。このパイプライン中で、IDは命令復号、AGはアドレス生成、CAはデータ・キャッシュ・アクセス、EXは命令実行を表す。命令のプット・アウェイ(put-away)は隠されている。すなわち、実行の結果は、次の実行サイクルで使用される必要な機能ユニットへの入力としてバイパスさせることができる。PUは、最大3個のESA/390命令を同時に実行できる能力をもつ設計になっているものとする。この能力により、2つのアドレス生成ALUがデー

20

*A/390命令でもオーバーラップは起こり得る。そうした命令としては、数式16が成立するときにオーバーラップが発生するPACKやZAP、数式17が成立するときにオーバーラップが発生するUNPK、および数式18が成立するときにオーバーラップが発生するTRとTRTがある

【数16】

$SAR_2 \leq SAR_1 + L_1 < SAR_2 + L_2$

【数17】

$SAR_2 + (2 * L_2) - 2$

タフロー中に含まれることになる。ここでは、MVCのようなSS命令では、MVC命令の復号に続くAGサイクル中にソース記憶アドレスと宛先記憶アドレスが同時に生成され得ることを示唆しておく。プロセッサのすべての命令はマイクロコードで制御され、単一サイクル命令は1つのマイクロ命令を呼び出し、複数サイクル命令はマイクロルーチンを呼び出す。MVCは複数サイクル命令の範囲に含まれ、したがってその実行中に複数のマイクロ命令が呼び出される。このようなルーチンでは、実行の結果、次のパイプライン・シーケンスが発生する。

第1 マイクロワード ID-AG-CA

第2 マイクロワード μF -AG-EX

第3 マイクロワード μF -AG-EX

上式で μF はマイクロワードの取出しを表し、命令の実行が完了するまでこのシーケンスが反復される。次に、本発明のオペランド・オーバーラップ予測装置を記述するのに使用するMVC命令の可能な実施態様について述べる。この実施態様では、第1のマイクロ命令が、第1オペランドでアドレス指定される記憶装置からのロードを指定する。この第1マイクロ命令のAGサイクル中に両方のオペランド・アドレスが2つのAGEN ALUによって計算され、オペランド1の取出しコマンドが形成され、データ・キャッシュに送られる。次のサイクルで、データ・キャッシュから取り出されたデータがデータ・バス上に置かれ、プロセッサに受諾される。プロセッサはキャッシュからデータ・バスに取り出されたデータをバイパスすることができるので、このサイクル中に取り出されたデータをデータ・キャッシュ中に格納するために記憶コマンドを形成することができると考えられる。しかし、アーキテクチャを遵守するには、データを記憶する前にオペランド・オーバーラップがわかっている必要がある。従来技術では、第2のマイクロ命令のAGサイクル中のオーバーラップを判定するため、第1命令のAGサイクル中に計算される記憶オペランド・アドレス SAR_1 と SAR_2 を次のサイクルでオーバーラップ検出論理機構にバイパスしなければならない。次にこのオーバーラップ検出の結果を分岐条件として使って、アーキテクチャに合致する形でMVCを実行するには2つのマイクロルーチンのどちらにアクセスすべきか

30

40

40

50

を判定する。その結果、従来技術では第2のマイクロ命令が実際にNOPとなり、取り出されたデータを格納するために適当なマイクロルーチンにアクセスする前に、オーバーラップを判定することが可能となる。したがって、記憶バスは過小利用となる。第1マイクロワードのAGサイクル中にオーバーラップ条件を予測し、第2のマイクロワードを取り出す際に適当な経路指定を選択するための分岐条件としてそれを利用できるようにすることができれば、この過小利用を軽減することができる。以下に、2つの記憶オペランド・アドレスSAR1とSAR2の計算と並行してオーバーラップ条件を予測できる、本発明を提示する。このようにすると、MVC型オーバーラップの検出を必要とする命令呼び出すごとに1サイクルが節約できる。代表的命令混合体を使用すると、オペランド長が8以下のMVCが呼び出される頻繁なケース(84%)ではMVC命令を実行するのに要するサイクル数が、平均4.7サイクルから平均3.7サイクルに減少し、21%の改善が得られる。MVCのオペランド長が8を越える比較的まれなケース(16%)では、実行サイクル数は11.8サイクルから10.8サイクルに減少し、9%の改善となる。CPIの純減少率は0.02であり、それが最小の追加ハードウェアで達成される。

【0028】オーバーラップをもたらすアドレス条件は5つのクラスに分けることができる。これらのケースを図1に示す。第1のケースでは、ソース・オペランドが高位記憶域と低位記憶域の境界を越えて折り返し(ラップ; wrap)、低位記憶域にある宛先オペランド・アドレスとオーバーラップするために、オーバーラップが発生する。第2のケースでは、ソース・オペランドと宛先オペランドが共に高位記憶域から低位記憶域に折り返す。この場合、宛先オペランドはソース・オペランドより上方にあり、低位アドレス位置に続いている。第3のケースでは、ソース・アドレスと宛先アドレスが記憶域内の同じ256バイト・ブロックで開始し、ソース・オペランドの全体がこの256バイト・ブロック内にある。第4のケースでは、両方のオペランドが記憶域の真中の同じ256バイト・ブロック内で開始し、両方のオペランドが共に次の順次256バイト・ブロック中にあふれ出る(スパイル)。最後に第5のケースでは、宛先オペランドがソース・オペランドの次の順次ブロックで開始する。ソース・オペランドはこの順次ブロック中にあふれ出して、宛先オペランドとオーバーラップする。オーバーラップしないケースは、以上のオーバーラップするケースに含まれない、残りのすべての場合を含む。これらのケースを図2および3に示す。次のこのオーバーラップするケースとオーバーラップしないケースのすべてを使って、ESA/390アーキテクチャでサポートされるすべてのアドレス指定モードでオーバーラップを検出するための本発明を開始する。このモードには、24ビット

および31ビット・アクセス・レジスタ・モードが含まれる。さらに、64ビットの基底と12ビットの変位のアドレス計算も仮定する。この24ビットと31ビットのアドレス指定モードが含まれるのは、それらがESA/390アーキテクチャで提供される24ビットおよび31ビット・アクセス・レジスタ・モードのサブセットと見なせるからである。図1のケース1やケースのような折返しのケースは、64ビットの基底と12ビットの変位の加算の場合、64ビット・アドレスのMSBが有効アドレスをもつためには0でなければならず、かつ折返しが $2^{64}-1$ アドレス境界で発生すると定義されているものと仮定することにより、考察から除外する。したがって、このアドレス指定モードでは、メモリ・アドレス指定が仮定アドレス空間の最高位256バイト・ブロックで発生し得ない。以下の考察では、2の補数を大文字、たとえばAで表すこととする。2の補数は、m個のビット a_i からなる。ただし、 $0 \leq i \leq m-1$ であり、 a_0 は最上位ビット(MSB)を表し、 a_{m-1} は最下位ビット(LSB)を表す。さらに、表記A($i:j$) (ただし、 $i \leq j \leq m-1$)を使って、A内部にあってiとjの間にあり両端を含むビットから構成される数を表すこととする。この表記を使用して、以下の定理を、MVC型オーバーラップを検出するための本発明の基礎として証明する。

【0029】定理1 2の補数演算を使って、2つの暗示的な正の2の補数の減算A-Bを行うと、 $A \geq B$ の場合、 $A=0 || A(i:m-1)$ および $B=0 || B(i:m-1)$ として、ビット位置iからすぐ下のビット位置*i-1*へのキャリーが発生する。

【0030】証明: $A(i:m-1) > B(i:m-1)$ の場合、 $a_j = 1$ かつ $b_j = 0$ であり、 $i \leq p < j$ であるすべてのpについて $a_p = b_p$ となるjが存在する。2の補数の減算A-Bを実行するには、Bの1の補数にAを加え、ホット1を位置m-1に与える。しかし、Bは1の補数化されているので、 $\bar{b}_p = \bar{a}_p$ かつ $a_j = 1$ および $\bar{b}_j = 1$ である。ただし、

【数21】

$\overline{b_p}$

を \bar{b}_p と表記する。他の文字にバーが付与される場合も、これに準ずる。したがって、 $A-B$ はjからj-1へのキャリーを発生する。しかし、位置j-1では、 $\bar{b}_j = \bar{a}_j$ であるため、 $a_{j-1} = 1$ かつ $\bar{b}_{j-1} = 0$ 、または $a_{j-1} = 0$ かつ $\bar{b}_{j-1} = 1$ である。したがって、j-1へのキャリーを a_{j-1} および \bar{b}_{j-1} に加えると、j-2へのキャリーが発生する。 $i+1 \leq q \leq j-2$ である各ビット位置qについてj-1の場合と同じ考察を適用すると、ビット位置iで、iへのキャリーは1であり、 a_i と b_i のどちらか一方だけが1となる。したがって、ビット位置iからi-1へのキャリーが発生することになる。

【 0 0 3 1 】 $A(i : m-1) = B(i : m-1)$ の場合、 $i \leq j \leq m-1$ であるすべての j について $a_j = b_j$ である。B は加算の前に補数化されるので、各ビット位置 j はすべて 0 に加算される 1 からなる。したがって ホット 1 が LSB 位置に与えられるので、 $m-1$ でキャリーが発生する。このキャリーが位置 $m-2$ に伝播し、そこで $m-3$ へのキャリーが発生する。このキャリーがあらゆるビット位置 j に伝播し、ビット i から $i-1$ へのキャリーが発生する。証明終り。

【 0 0 3 2 】 定理 2-2 の補数演算を使って、2 つの暗示的な正の 2 の補数の減算 $A - B$ を行うと、 $A < B$ の場合、 $A = 0 || A(i : m-1)$ および $B = 0 || B(i : m-1)$ として、ビット位置 i から次のビット位置 $i-1$ へのキャリーは発生しない。
10

【 0 0 3 3 】 証明: $A(i : m-1) < B(i : m-1)$ の場合、 $a_j = 0$ かつ $b_j = 1$ であり、 $i \leq p < j$ となるすべての p について $a_p = b_p$ となる j が存在する。2 の補数の減算 $A - B$ を実行するには、B の 1 の補数に A を加え、ホット 1 を位置 $m-1$ に与える。しかし、B は 1 の補数化されているので、 $b_p = \bar{a}_p$ かつ $a_j = 0$ および $\bar{b}_j = 0$ である。 a_j と \bar{b}_j が共に 0 なので、位置 j へのキャリーである c_{j+1} の値がどうであれ、 j から $j-1$ へのキャリーは 0 となる。したがって、 $A - B$ は、 j から $j-1$ へのキャリーが発生しない。位置 $j-1$ では、 $\bar{b}_j = \bar{a}_{j-1}$ であるため、 $a_{j-1} = 1$ かつ $\bar{b}_{j-1} = 0$ または $a_{j-1} = 0$ かつ $\bar{b}_{j-1} = 1$ である。したがって、 $j-1$ へのキャリーがないことは、 a_{j-1} と \bar{b}_{j-1} を加算して $j-1$ へのキャリー $-c_{j-1}$ が発生しても、 $j-2$ へのキャリーは発生し

ないことを意味している。 $i+1 \leq q \leq j-2$ である各ビット位置 q について $j-1$ の場合と同じ考察を適用すると、ビット位置 i で、 i へのキャリーは 0 であり、 a_i と \bar{b}_i のどちらか一方だけが 1 となる。したがって、ビット i から $i-1$ へのキャリーは発生しない。証明終り。

【 0 0 3 4 】 MVC 型オペランド・オーバーラップを検出するための本発明を引き続き展開するため、以下の表記法を利用する。まず、SAR2 と SAR1 をそれぞれ A と B で表す。両者はそれぞれ a_i と b_i で表される個々のビットを有する。ただし、 $0 \leq i \leq 63$ である。8 ビット・オペランド長を OL(0 : 7) で表し、左に 56 個の 0 を連結したものを L で表す。したがって、L は、 $0 \leq i \leq 55$ の場合は $0, 56 \leq i \leq 63$ の場合は OL(0 : 7) となる。以下の議論では、A、B、L の以下の加算／減算の組合せが必要となる。その組合せならびにその表記法は、

1. $A - B$ これを Δ で表し、個々のビットを δ_i で表し、あるビット位置から次のビット位置へのキャリーを κ_i で表す。
2. $A + L$ これを E で表し、個々のビットを ϵ_i で表し、キャリーを γ_i で表す。
3. $\Delta + L$ これを S で表し、個々のビットを S_i で表し、キャリーを c_i で表す。
4. $E - B$ これを S^+ で表し、個々のビットを S_i^+ で表し、キャリーを λ_i で表す。

この表記法を表 1 に要約して示す。

【 表 1 】

MVC 型オーバーラップ検出装置用の表記法				
エンティティ	表現	文字表現	ビット表現	キャリー表現
SAR2	-	A	a_i	-
SAR1	-	B	b_i	-
SAR2 - SAR1	$A - B$	Δ	δ_i	κ_i
SAR2 + L	$A + L$	E	ϵ_i	γ_i
$(SAR2 - SAR1) + L$	$\Delta + L$	S	S_i	c_i
$(SAR2 + L) - SAR1$	$E - B$	S^+	S_i^+	λ_i

21

【0035】上記の表記法を用いると、以下の定理が成立する。

【0036】定理3 記憶オペランド1および2がMV C型オーバーラップを有する場合、SAR2がアドレス空間の高位256バイト・ブロック内にあり、SAR1が低位256バイト・ブロック内にあるときに、SAR1がSAR2より上にマップされるならば、 $0 \leq i \leq 5$ *

22

*5の場合 $\delta_i = 1$ である。

【0037】証明: ESA/390アーキテクチャで提供される3つのアドレス指定モードのそれぞれについて、図1に示した各ケースごとにこの定理を証明しなければならない。各アドレス指定モードの場合の記憶アドレスの表現を表2に示す。

【表2】

ESA/390および想定した64ビット・アドレス指定モードでのアドレス表現	
モード	アドレス表現
24ビット・アクセス・レジスタ (AR24ビット) モード	$a_0 a_1 a_{31} 0 0 \dots 0 a_{40} a_{41}$ $\dots a_{55} a_{56} a_{57} \dots a_{63}$
31ビット・アクセス・レジスタ (AR31ビット) モード	$a_0 a_1 \dots a_{31} 0 a_{33} \dots a_{55} a_{56}$ $a_{57} \dots a_{63}$
64ビット・アドレス指定	$0 a_1 \dots a_{55} a_{56} a_{57} \dots a_{63}$

【0038】表2では、0であることがわかっているSAR2のビット位置は0で示してあり、1と0のどちらかであるビット位置は、SAR2を先にAで表したので添字付きのaで表してある。添字は変数中のビット位置を表す。同様に、aの代りにbを使ってSAR1を表すことができる。この表現を使って、図1に示した各ケースについて考察しなければならない。SAR2がアドレス空間の高位256バイト・ブロック内にあり、SAR1が低位256バイト・ブロック内にある場合では、アドレスを1ビット位置拡張し、そのビット位置を1にすることにより、SAR1をSAR2より上にマップすることができる。したがって、24ビット・アドレス空間の場合、そのアドレスのMSBに25番目のビットが連結され、それが1にされることになる。表2に示したアドレス表現では、各アドレス指定モードについてアドレス空間がすでに拡張されており、表に示すようにビット位置が0である。したがって、この表現でSAR1をSAR2より上にマップするには、次のビット位置を1にするだけでよい。たとえば、アドレスの24ビットは、アドレスの少なくとも24ビット位置を必要とするはずであり、これは表2の表現ではビット40~63に対応する。24ビット・アドレス指定では、SAR1をSAR2より上にマップするために、ビット39を0にする。

同様に31ビット・アドレス指定では、SAR1をSAR2より上にマップするため、アドレスのビット32を1にする。

【0039】定理3 ケース1 ケース1は、Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあり、Aの最上位ビットがBの最上位ビットに等しいことを特徴とする。Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあるため、BはAより上にマップされるはずである。さらに、 $B(56:63) \leq E(56:63)$ かつ $E(56:63) < A(56:63)$ なので、 $A(56:63) > B(56:63)$ である。したがって、定理1により、 $\kappa_{56}=1$ となる。以下で3つのアドレス指定モードのすべてを考慮する。

【0040】AR24ビット・モード このアドレス指定モードでは、BをAより上にマップするため、ビット39を論理1にする。ビット39より上位のビット位置は等しいので、 $i \leq 38$ であるiについて $a_i = b_i$ である。その結果、A-Bは数式22で表すことができ、ホット1が暗黙のうちに強制的にビット63に入れられる。

【数22】

23

 $a_0 \ a_1 \dots a_{31} 00 \dots 011 \dots 1 a_{56} b_{57} \dots b_{63}$

24

 $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 11 \dots 011 \dots 1 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0041 】 $\kappa_{56}=1$ であり、かつ $40 \leq i \leq 55$ である各ビット位置がその位置へのキャリーに加算すべき 1 を 2 個有するので、 $40 \leq i \leq 55$ であるすべての i について、 $\delta_i=1$ かつ $\kappa_i=1$ となる。ビット位置 39 では 2 個の 0 が $\kappa_{40}=1$ と加算され、したがって $a_{39}=1$ かつ $\kappa_{39}=0$ である。 $0 \leq i \leq 38$ である各ビット位置では、1 とキャリーの 0 に 0 が加算される。したがって、 $0 \leq i \leq 38$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【 0042 】 AR31 ビット・モード このアドレス指定モードでは、B を A より上にマップするために、ビット 32 を論理 1 にする。ビット 32 より上位のビット位置は等しいので、 $i \leq 31$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式 23 で表すことができ、ホット 1 が暗黙のうちに強制的にビット位置 63 に入れられる。

【 数23 】

 $a_0 \ a_1 \dots a_{31} 01 \dots 1 a_{56} b_{57} \dots b_{63}$ $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 01 \dots 1 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0043 】 24 ビット・アドレス指定の場合と同様に、 $\kappa_{56}=1$ であり、かつ $33 \leq i \leq 55$ である各ビット

 $a_0 \ a_1 \dots a_{31} 00 \dots 011 \dots 1 a_{56} b_{57} \dots b_{63}$ $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 11 \dots 100 \dots 0 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0047 】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき 1 個の 1 と 1 個の 0 を有するので、各ビットでの結果は、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【 0048 】 AR31 ビット・モード ビット 33 より上位のビット位置が B と A で等しく、したがって $i \leq 32$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式 25 であらわすことができ、ホット 1 が暗黙のうちに強制的にビット位置 63 に入れられる。

【 数25 】

 $a_0 \ a_1 \dots a_{31} 01 \dots 1 a_{56} b_{57} \dots b_{63}$ $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 10 \dots 0 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0049 】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が 0 及びそのビット位置へのキャリーに加

*ト位置がその位置へのキャリーに加算すべき 1 を 2 個有するので、 $33 \leq i \leq 55$ であるすべての i について、 $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 32 では 2 個の 0 が $\kappa_{33}=1$ と加算され、したがって $\delta_{32}=1$ かつ $\kappa_{32}=0$ である。 $0 \leq i \leq 31$ である各ビット位置については、1 とキャリーの 0 に 0 が加算される。したがって、 $0 \leq i \leq 31$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【 0044 】 64 ビット・アドレス指定 64 ビット・アドレス指定ではケース 1 は生じ得ない。その結果、64 ビット・アドレス指定の場合は、このケースに対処するために B を A より上にマップする必要はない。

【 0045 】 定理 3 ケース 2 ケース 2 は、A と B が共にアドレス空間の高位 256 バイト・ブロック内にあり、 $A(56:63) < B(56:63)$ であることを特徴とする。したがって、定理 2 より、 $\kappa_{56}=0$ となる。

【 0046 】 AR24 ビット・モード ビット 40 より上位のビット位置が B と A で等しく、したがって $i \leq 39$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式 24 で表すことができ、ホット 1 が暗黙のうちに強制的にビット位置 63 に入れられる。

【 数24 】

 $a_0 \ a_1 \dots a_{31} 00 \dots 011 \dots 1 a_{56} b_{57} \dots b_{63}$

算すべき 1 個の 1 を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【 0050 】 64 ビット・アドレス指定 64 ビット・アドレス指定ではケース 2 は生じ得ない。

【 0051 】 定理 3 ケース 3 および 4 これらのケースは、B と A が共にアドレス空間の同一の 256 バイト・ブロック内にあり、 $A(56:63) < B(56:63)$ であることを特徴とする。定理 2 より、 $\kappa_{56}=0$ となる。B と A の両方が存在する 256 バイト・ブロックは任意である。

【 0052 】 AR24 ビット・モード ビット 40 より上位のビット位置が B と A で等しく、したがって $i \leq 39$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式 26 で表すことができ、ホット 1 が暗黙のうちに強制的にビット位置 63 に入れられる。

【 数26 】

25

 $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ a_{40} \ a_{41} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$

26

 $\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ 1 \dots 1 \ \overline{a}_{40} \ \overline{a}_{41} \dots \overline{a}_{55} \ \overline{a}_{56} \ \overline{a}_{57} \dots \overline{b}_{63}$

【0053】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0 及びそのビット位置へのキャリヤーに加算すべき1 個の1 を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべてのi について $\delta_i=1$ である。

【0054】 AR31ビット・モード ビット33より*

 $a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ \overline{a}_{33} \dots \overline{a}_{55} \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0055】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0 及びそのビット位置へのキャリヤーに加算すべき1 個の1 を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべてのi について $\delta_i=1$ である。

【0056】 64ビット・アドレス指定 このケースで 20 は64ビット・アドレス指定の場合、A-Bは次式で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数28】

 $0 \ a_1 \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$
 $1 \ \overline{a}_1 \dots \overline{a}_{55} \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0057】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0 及びそのビット位置へのキャリヤーと加算すべき1 個の1 を有するので、 $\delta_i=0$ かつ $\kappa_i=0$ で※ 30

 $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ a_{40} \ a_{41} \dots a_{j-1} \ 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ 1 \dots 1 \ \overline{a}_{40} \ \overline{a}_{41} \dots \overline{a}_{j-1} \ 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0060】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーと加算すべき1を2個有するので、 $j+1 \leq i \leq 55$ あるiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置jでは a_j と b_j が共に0 であり、かつ $\kappa_{j+1}=1$ である。したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間の各ビット位置では、加数のうちただ1個が1 であり、そのビット位置へのキャリヤーは0 である。したがって、 $0 \leq i \leq j-1$ であるiについて $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべてのiについて $\delta_i=1$ で★

 $a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{j-1} \ 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ \overline{a}_{33} \dots \overline{a}_{j-1} \ 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0062】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ で 50 ある各ビット位置がそのビット位置へのキャリヤーと加算

*上位のビット位置がBとAで等しく、したがって $i \leq 3$ 2 であるiについて $a_i=b_i$ である。その結果、A-Bは数式27で表すことができ、ホット1が暗黙のうちに強制的にビット63に入れられる。

【数27】

※ある。したがって、 $0 \leq i \leq 55$ であるすべてのiについて $\delta_i=1$ である。

【0058】 定理3 ケース5 このケースでは、Bは、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にある。さらに、B(56:6) $\leq E(56:63)$ かつ $E(56:63) < A(56:63)$ である。したがって、A(56:63) $> B(56:63)$ なので、定理1により $\kappa_{56}=1$ である。

【0059】 AR24ビット・モード Bが、Aの存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $40 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ であるpについて $a_p=1$ かつ $b_p=0$ となる位置j が存在する。さらに、 $i < j$ の場合、 $a_i=b_i$ である。その結果、A-Bは数式29で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数29】

 $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ a_{40} \ a_{41} \dots a_{j-1} \ 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

★ある。

【0061】 AR31ビット・モード Bが、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $33 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ であるpについて $a_p=1$ かつ $b_p=0$ となる位置j が存在する。さらに、 $i < j$ の場合、 $a_i=b_i$ である。その結果、A-Bは数式30で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数30】

 $a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{j-1} \ 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ \overline{a}_{33} \dots \overline{a}_{j-1} \ 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

すべき1を2個有するので、 $j+1 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ かつ $\kappa_i = 1$ である。ビット位置*j*では、 a_j と b_j が共に0であり、 $\kappa_{j+1} = 1$ である。したがって、 $\delta_j = 1$ かつ $\kappa_j = 0$ である。0と*j*-1の間の各ビット位置では、加数のうちの1個だけが1であり、そのビット位置へのキャリーは0である。したがって $0 \leq i \leq j-1$ である*i*について、 $\delta_i = 1$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ である。

* $0 \ a_1 \dots a_{j-1} \ 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$1 \ \overline{a}_1 \dots \overline{a}_{j-1} \ 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0064】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、 $j+1 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ かつ $\kappa_i = 1$ である。ビット位置*j*では、 $a_j = b_j$ が共に0であり、 $\kappa_{j+1} = 1$ である。したがって、 $\delta_j = 1$ かつ $\kappa_j = 0$ である。0と*j*-1の間の各ビット位置では、加数のうちの1個だけが1であり、そのビット位置へのキャリーは0である。したがって、 $0 \leq i \leq j-1$ である*i*について $\delta_i = 1$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ である。証明終り。

【0065】定理4 記憶オペランド1及び2がMVC型オーバーラップを有せず、SAR2がアドレス空間の高位256バイト・ブロック内にありSAR1が低位256バイト・ブロック内にあるときSAR1がSAR2より上にマップされる場合、減算SAR2-SAR1の結果 δ_i は、非オーバーラップ・ケース1、6、11、ビット56からビット55へのキャリーが生成される($\kappa_{56}=1$)とき、及び非オーバーラップ・ケース3を除き $0 \leq i \leq 55$ の場合すべて1ではない。

【0066】定理4 ケース1

* $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ 1 \dots 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

【0068】サブケース1 ビット位置55について考える。ビット位置55は $\kappa_{56}=0$ に加算すべき1を2個有するので、 $\delta_{55}=0$ である。したがって、このサブケースでは、 $0 \leq i \leq 55$ である*i*について δ_i がすべて1ではない。

【0069】サブケース2 $\kappa_{56}=1$ であり、かつ $40 \leq i \leq 55$ である各ビット位置がその位置へのキャリーに加算すべき1を2個有するので、 $40 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ かつ $\kappa_i = 1$ である。ビット位置39では、 $\kappa_{40}=1$ に0が2個加算される。したがって $\delta_{39}=1$ かつ $\kappa_{39}=0$ である。 $0 \leq i \leq 38$ である各ビット位置では、1とそのビット位置へのキャリー

*【0063】64ビット・アドレス指定 Bが、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $1 \leq j \leq 55$ であって、 $a_j = 0$ かつ $b_j = 1$ であり、 $j < p \leq 55$ である*p*について $a_p = 1$ かつ $b_p = 0$ となる位置*j*が存在する。さらに、 $i < j$ の場合、 $a_i = b_i$ である。その結果、A-Bは数式31で表すことができ、ホット1が暗黙のうちに強制的に位置63に入れられる。

【数31】

※ケース1は、Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあり、Aの最上位ビットがBの最上位ビットと等しいことを特徴とする。Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあるので、BはAより上にマップしなければならない。さらに、A(56:63)がB(56:63)未満である可能性があり、その場合は定理2により $\kappa_{56}=0$ (サブケース1)である。また、A(56:63)がB(56:63)以上である可能性もあり、その場合は定理1により $\kappa_{56}=1$ (サブケース2)である。3つのアドレス指定モードのすべてについてこの両方のサブケースを考慮しなければならない。

【0067】AR24ビット・モード このアドレス指定モードでは、BをAより上にマップするために、ビット39を強制的に論理1にする。ビット39より上位のビット位置は等しいので、 $i \leq 38$ である*i*について $a_i = b_i$ である。その結果、A-Bは数式32で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数32】

* $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$\overline{a}_0 \ \overline{a}_1 \dots \overline{a}_{31} \ 1 \ 1 \dots 0 \ 1 \ 1 \dots 1 \ \overline{b}_{56} \ \overline{b}_{57} \dots \overline{b}_{63}$

とし0が加算される。 $\kappa_{39}=0$ なので、 $0 \leq i \leq 38$ であるすべての*i*について $\delta_i = 1$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ である。

【0070】AR31ビット・モード このアドレス指定モードでは、BをAより上にマップするために、ビット32を強制的に論理1にする。ビット32より上位のビット位置は等しいので、 $i \leq 31$ である*i*について $a_i = b_i$ である。その結果、A-Bは数式33で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数33】

29

 $a_0 \ a_1 \dots a_{31} \ 0 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 0 \ 1 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0 0 7 1 】 サブケース1 ビット位置5 5について考
える。ビット位置5 5は、 $\kappa_{56}=0$ と加算すべき1を2
個有するので、 $\delta_{55}=0$ である。したがってこのサブケ
ースでは、 $0 \leq i \leq 5 5$ であるiについて δ_i はすべて
1ではない。

【 0 0 7 2 】 サブケース2 $\kappa_{56}=1$ であり、かつ $3 \ 3 \leq i \leq 5 \ 5$ である各ビット位置がその位置へのキャリーと加算すべき1を2個有するので、 $3 \ 3 \leq i \leq 5 \ 5$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置3 2では、 $\kappa_{32}=1$ に2個の0が加算される。したがって、 $\delta_{32}=1$ かつ $\kappa_{32}=0$ である。 $0 \leq i \leq 3 \ 1$ である各ビット位置について、1とそのビット位置へのキャリーに0が加算される。 $\kappa_{32}=0$ であるので、 $0 \leq i \leq 3 \ 1$ であるすべてのiについて $\delta_i=1$ であり、したがって $0 \leq i \leq 5 \ 5$ であるすべてのiについて $\delta_i=1$ である。

10

*【 0 0 7 3 】 6 4ビット・アドレス指定 6 4ビット・アドレス指定ではケース1は起こり得ない。したがって、このケースの6 4ビット・アドレス指定モードではBをAより上にマップする必要はない。証明終り。

【 0 0 7 4 】 定理4 ケース2 ケース2は、BとAが共にアドレス空間の最高位2 5 6 バイト・ブロック内にあり、 $A(56:63) \geq B(56:63)$ であることを特徴とする。したがって、定理1により $\kappa_{56}=1$ である。

【 0 0 7 5 】 AR 2 4ビット・モード ビット4 0より上位のビット位置はBとAで等しいので、 $i \leq 3 \ 9$ であるiについて $a_i=b_i$ である。その結果、 $A-B$ は数式3 4で表すことができ、ホット1が暗黙のうちに強制的にビット位置6 3に入れられる。

【 数3 4 】

*20

 $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ 1 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 1 \ 1 \dots 1 \ 0 \ 0 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0 0 7 6 】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 5 \ 5$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、 $0 \leq i \leq 5 \ 5$ であるすべてのiについて $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 5 \ 5$ であるすべてのiについて $\delta_i \neq 1$ である。

※30

 $a_0 \ a_1 \dots a_{31} \ 0 \ 1 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 1 \ 0 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0 0 7 8 】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 5 \ 5$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、 $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 5 \ 5$ であるすべてのiについて $\delta_i \neq 1$ である。

【 0 0 7 9 】 6 4ビット・アドレス指定 6 4ビット・アドレス指定ではケース2は起こり得ない。したがって、これを扱う必要はない。

【 0 0 8 0 】 定理4 ケース3 ケース3は、BとAがアドレス空間の同じ2 5 6 バイト・ブロック内にあり、★

*【 0 0 7 7 】 AR 3 1ビット・モード ビット3 3より上位のビット位置はBとAで等しいので、 $i \leq 3 \ 2$ であるiについて $a_i=b_i$ である。その結果、 $A-B$ は数式3 5で表すことができる。ホット1が暗黙のうちに強制的にビット位置6 3に入れられる。

【 数3 5 】

★ $B(56:63) > E(56:63) > A(56:63)$ であることを特徴とする。すなわち、 $A(56:63) < B(56:63)$ であるので、定理2により $\kappa_{56}=0$ である。2 5 6 バイト・ブロックは任意である。

【 0 0 8 1 】 AR 2 4ビット・モード ビット4 0より上位のビット位置は等しいので、 $i \leq 3 \ 9$ であるiについて $a_i=b_i$ である。その結果、 $A-B$ は数式3 6で表すことができる。ホット1が暗黙のうちに強制的にビット位置6 3に入れられる。

【 数3 6 】

 $a_0 \ a_1 \dots a_{31} \ 0 \ 0 \dots 0 \ a_{40} \ a_{41} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$
 $\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 1 \ 1 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【 0 0 8 2 】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 5 \ 5$ である各ビット位置がそのビット位置へのキャリーと加算すべ
き1を1個と0を1個有するので、各ビット位置につ
いて $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 5$

き1を1個と0を1個有するので、各ビット位置につ
いて $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 5$

50

31

5 であるすべての i について $\delta_i = 1$ である。

【 0 0 8 3 】 AR31ビット・モード ビット33より上位のビット位置が等しいので、 $i \leq 32$ である i について $a_i = b_i$ である。その結果、A-Bは数式37で表*

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 0 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{a_{56}} \ \overline{a_{57}} \dots \overline{a_{63}}$$

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【 0 0 8 4 】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i = 1$ かつ $\kappa_i = 0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i = 1$ である。

【 0 0 8 5 】 64ビット・アドレス指定 64ビット・アドレス指定モードではA-Bは数式38で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【 数38 】

$$0 \ a_1 \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$$

$$1 \ \overline{a_1} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【 0 0 8 6 】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i = 1$ かつ $\kappa_i = 0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i = 1$ である。

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 0 \ 0 \dots 0 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{55}} \ \overline{a_{56}} \ \overline{a_{57}} \dots \overline{a_{63}}$$

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 1 \ 1 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【 0 0 9 0 】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーに加算すべき0を1個と1を1個有するので、各ビット位置について $\delta_i = 0$ かつ $\kappa_i = 1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【 0 0 9 1 】 AR31ビット・モード ビット33より★

$$a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【 0 0 9 2 】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i = 0$ かつ $\kappa_i = 1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【 0 0 9 3 】 64ビット・アドレス指定 64ビット・アドレス指定モードではA-Bは数式41で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【 数41 】

* すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【 数37 】

※き1を1個と0を1個有するので、各ビット位置について $\delta_i = 0$ かつ $\kappa_i = 0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i = 0$ である。

【 0 0 8 7 】 ケース3について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i = 1$ である。

【 0 0 8 8 】 定理48ケース4及び5 この2つのケースは、BとAがアドレス空間の同じ256バイト・ブロック内にあり、A(56:63) ≥ B(56:63) であることを特徴とする。したがって、定理により $\kappa_{56}=1$ である。256バイト・ブロックは任意である。

【 0 0 8 9 】 AR24ビット・モード ビット40より上位のビット位置は等しいので、 $i \leq 39$ である i について $a_i = b_i$ である。その結果、A-Bは次のように表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【 数39 】

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 0 \ 0 \dots 0 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{55}} \ \overline{a_{56}} \ \overline{a_{57}} \dots \overline{a_{63}}$$

★上位のビット位置は等しいので、 $i \leq 32$ である i について $a_i = b_i$ である。その結果、A-Bは数式40で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【 数40 】

$$a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

$$0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \dots a_{63}$$

$$1 \ \overline{a_1} \dots \overline{a_{31}} \ \overline{b_{33}} \dots \overline{b_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【 0 0 9 4 】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリヤーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i = 0$ かつ $\kappa_i = 1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【 0 0 9 5 】 ケース4と5について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【 0 0 9 6 】 定理4 ケース6及び11 この2つのケ

33

ースでは、Aのある256バイト・ブロックのすぐ上の256バイト・ブロックにBがある。さらに、A(5:63)がB(56:63)未満である可能性があり、その場合は定理2から $\kappa_{56}=0$ である(サブケース1)。またA(56:63)がB(56:63)以上である可能性もあり、その場合は定理1から $\kappa_{56}=1$ である(サブケース2)。3つのアドレス指定モードのそれについて両方のサブケースを考慮しなければならない。

$a_0 \ a_1 \dots a_{31} \ 00 \dots 0 \ a_{40} \ a_{41} \dots a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 11 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【0098】サブケース1 ビット位置55について考える。この位置では、2個の1が0である κ_{56} に加算される。その結果 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべてのiについて $\delta_{55}=0$ である。

【0099】サブケース2 $\kappa_{56}=1$ であり、かつ($j+1$) $\leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を2個有するので、($j+1$) $\leq i \leq 55$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置jでは、2個の0が $\kappa_{j+1}=1$ に加算され、したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間にあるすべてのビットで1個の1と1個の0がそのビット位置へのキャリーに加算される。※

$a_0 \ a_1 \dots a_{31} \ 0 \ a_{33} \dots a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【0101】サブケース1 ビット位置55について考える。2個の1が $\kappa_{56}=0$ に加算される。その結果、 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべてのiについて $\delta_i \neq 1$ である。

【0102】サブケース2 $\kappa_{56}=1$ であり、かつ($j+1$) $\leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を2個有するので、($j+1$) $\leq i \leq 55$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置jでは、2個の0が $\kappa_{j+1}=1$ に加算され、したがって $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間にある各ビット位置では、1個の0と1個の1がそのビット位置へのキャリーに加算される。 $\kappa \star$

$0 \ a_1 \dots a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \dots a_{63}$

$1 \ \overline{a_1} \dots \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$

【0104】サブケース1 ビット位置55について考える。この位置では、2個の1が $\kappa_{56}=0$ に加算される。その結果、 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべてのiについて $\delta_i \neq 1$ である。

【0105】サブケース2 $\kappa_{56}=1$ であり、かつ($j+1$) $\leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、($j+1$) $\leq i \leq 55$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置jでは、2個の0が $\kappa_{j+1}=1$ に加算され、したがって、 $\delta_j=1$ かつ $\kappa_j=0$ であ

34

*【0097】AR24ビット・モード Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $40 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ であるpについて $a_p=1$ かつ $b_p=0$ である位置jが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは次式のように表すことができる。ホット1が暗黙のうちにビット位置63に強制される。

*【数42】

※ $\kappa_j=0$ であるので、 $0 \leq i \leq (j-1)$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるiについて $\delta_i=1$ である。

【0100】AR31ビット・アドレス指定 Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $33 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ であるpについて $a_p=1$ かつ $b_p=0$ となるjが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは数式43で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

*【数43】

※ $\kappa_j=0$ であるので、 $0 \leq i \leq (j-1)$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるiについて $\delta_i=1$ である。

【0103】64ビット・アドレス指定 Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $0 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ であるpについて $a_p=1$ かつ $b_p=0$ となるjが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは数式44で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

*【数44】

+1) $\leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、($j+1$) $\leq i \leq 55$ であるすべてのiについて $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置jでは、2個の0が $\kappa_{j+1}=1$ に加算され、したがって、 $\delta_j=1$ かつ $\kappa_j=0$ であ

る。0と $j-1$ の間にある各ビット位置では、1個の1と1個の0がそのビット位置へのキャリヤーに加算される。 $\kappa_j=1$ であるので、 $0 \leq i \leq j-1$ であるすべての*i*について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ である*i*について $\delta_i=1$ である。

【0106】ケース6及び11について要約すると、 $\kappa_{56}=0$ の場合、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。 $\kappa_{56}=1$ の場合は、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ である。

【0107】定理4 ケース7 このケースでは、Bがある256バイト・ブロックのすぐ上の256バイト・ブロックにAがある。A(56:63)はB(56:63)未満である可能性があり、その場合は定理2から $\kappa_{56}=0$ である(サブケース1)。またA(56:63) *

$a_0 a_1 \dots a_{31} 00 \dots 0 a_{40} a_{41} \dots a_{j-1} 100 \dots 0 a_{56} a_{57} \dots a_{63}$

$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} 11 \dots 1 \overline{a_{40}} \overline{a_{41}} \dots \overline{a_{j-1}} 100 \dots 0 \overline{b_{56}} \overline{b_{57}} \dots \overline{b_{63}}$

【0109】ビット位置jについて考える。この位置では、2個の1がそのビット位置へのキャリヤーに加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置j-1では、1個の1と1個のゼロが κ_j に加算される。したがって、 $\kappa_j=1$ であるので $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0110】31ビット・アドレス指定のアクセス・レジスターBがある256バイト・ブロックのすぐ上の2*

$a_0 a_1 \dots a_{31} 0 a_{33} \dots a_{j-1} 100 \dots 0 a_{56} a_{57} \dots a_{63}$

$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} 1 \overline{a_{33}} \dots \overline{a_{j-1}} 100 \dots 0 \overline{b_{56}} \overline{b_{57}} \dots \overline{b_{63}}$

【0111】ビット位置jについて考える。この位置では、2個の1がそのビット位置へのキャリヤーと加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置j-1では、1個の1と1個の0が κ_j に加算される。したがって、 $\kappa_j=1$ である $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0112】64ビット・アドレス指定 Bのある25★
0 a₁ ... a_{j-1} 100 ... 0 a₅₆ a₅₇ ... a₆₃

1 a₁ ... a_{j-1} 100 ... 0 b₅₆ b₅₇ ... b₆₃

【0113】ビット位置jについて考える。この位置では、2個の1がそのビット位置へのキャリヤーと加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置j-1では、1個の1と1個の0が κ_j に加算される。したがって、 $\kappa_j=1$ であるので $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

*がB(56:63)以上である可能性もあり、その場合は定理1から $\kappa_{56}=1$ である(サブケース2)。各アドレス指定モードについて両方のサブケースを同時に考慮することができる。

【0108】AR24ビット・モード Bがある256バイト・ブロックのすぐ上の256バイト・ブロックにAがあるので、 $40 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ であるpについて $a_p=0$ かつ $b_p=1$ となるjが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは数式45で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数45】

20 *56バイト・ブロックにAがあるので、 $33 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ であるpについて $a_p=0$ かつ $b_p=1$ となる位置jが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは数式46で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数46】

$a_0 a_1 \dots a_{31} 0 a_{33} \dots a_{j-1} 100 \dots 0 a_{56} a_{57} \dots a_{63}$

$\overline{a_0} \overline{a_1} \dots \overline{a_{31}} 1 \overline{a_{33}} \dots \overline{a_{j-1}} 100 \dots 0 \overline{b_{56}} \overline{b_{57}} \dots \overline{b_{63}}$

★6バイト・ブロックのすぐ上の256バイト・ブロックにAがあるので、 $1 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ であるpについて $a_p=0$ かつ $b_p=1$ となる位置jが存在する。さらに、 $i < j$ であるiについて $a_i=b_i$ である。その結果、A-Bは数式47で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数47】

$0 a_1 \dots a_{j-1} 100 \dots 0 a_{56} a_{57} \dots a_{63}$

$1 \overline{a_1} \dots \overline{a_{j-1}} 100 \dots 0 \overline{b_{56}} \overline{b_{57}} \dots \overline{b_{63}}$

【0114】ケース7について要約すると、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0115】定理4 ケース8、9、10、12 これらのケースでは、AとBのある256バイト・ブロックが少なくとも1個のバイト・ブロックで分離されている。これらのケースでは、すべてのアドレス・モードを同じに扱うことができる。これらのケースでは、ビット

37

位置1と55の間に、 $i < j$ であるすべての*i*について $a_i \neq b_i$ かつ $a_i = b_i$ であるビット位置 j が少なくとも1つ存在する。

【0116】まず $a_j = \bar{b}_j = 1$ のときのビット位置 j について考える。この条件のときは、 $\kappa_{j+1} = 0$ であろうと $\kappa_{j+1} = 1$ であろうと、 $\kappa_j = 1$ である。位置 $j-1$ では、1個の1と1個の0が κ_j に加算されて $\delta_{j-1} = 0$ となり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0117】次に、 $a_j = \bar{b}_j = 0$ かつ $\kappa_{j+1} = 0$ という条件について考える。 a_j 、 b_j および κ_{j+1} の加算の結果は $\delta_j = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0118】最後に、 $a_j = \bar{b}_j = 0$ かつ $\kappa_{j+1} = 1$ の場合について考える。この場合は $\delta_j = 1$ かつ $\kappa_j = 0$ である。 $i < j$ である *i*について $\bar{b}_i = \bar{a}_i$ であり、かつ $\kappa_i = 0$ なので、 $0 \leq i < j$ であるすべての*i*について $\delta_i = 1$ かつ $\kappa_i = 0$ である。しかし $\kappa_{j+1} = 1$ の場合は、 $j+1$ の両方の加数が1であり、 $\kappa_{j+2} = 0$ である（サブケース1）か、または一方の加数が1、他方が0であり、 $\kappa_{j+2} = 1$ である（サブケース2）であるか、または両方の加数が1であり、 $\kappa_{j+2} = 1$ である（サブケース3）。サブケース1と2では、 $\delta_{j+1} = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。しかし、サブケース3では、 $\kappa_{j+2} = 1$ であるので $\delta_{j+1} = 1$ である。ビット位置 $j+2$ でも同じ3つのサブケースが生じる。したがって、 $\delta_{j+2} = 1$ の場合、 $j+2$ の両方の加数が1で $\kappa_{j+3} = 1$ でなければならず、あるいは $\delta_{j+2} = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。 $(j+1) \leq p \leq 55$ である *p*について同様の議論がビット毎に繰り返され、したがって $(j+1) \leq p \leq 55$ であるすべての *p*について $\delta_p = 1$ であり、したがって $a_p = \bar{b}_p = 1$ である。したがって、 $a_p = 1$ かつ $b_p = 0$ であり、その結果、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i = 1$ ならばAが存在する。

6バイト・ブロックのすぐ上の256バイト・ブロックにBが存在しなければならなくなる。しかし、これはBとAが1つ以上の256バイト・ブロックで分離されているというこのケースに対する仮定と矛盾する。その結果、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0119】ケース8、9、10、12について要約すると、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0120】12のケースのそれぞれの考察からわかるように、記憶オペランド1及び2がMVC型のオーバーラップを含まない場合、 $\kappa_{56} = 1$ のときの非オーバーラップ・ケース1、6、11及び非オーバーラップ・ケース3を除き、 $0 \leq i \leq 55$ であるすべての*i*について δ

38

$i \neq 1$ である。証明終り。

【0121】定理5 $0 \leq i \leq 55$ であるすべての*i*について減算SAR2-SAR1の結果である δ_i が論理1であるなら、加算(SAR2-SAR1)+Lの間に生成されるビット位置56からビット位置55へのキャリーは、記憶オペランドがMVC型オーバーラップを含む場合、1である。

【0122】証明

i_f の証明

加算の累加性から、SAR2(55:63)-SAR1(55:63)+L(55:63)は、 $\Delta(55:63)+L(55:63)$ またはE(55:63)-B(55:63)のどちらかによって生成できる。その結果、式48が成立する。

【数48】

$$S_{55} = S_{55}^\dagger$$

【0123】しかし、式49が成立し、 $L_{55}=0$ なので式50が成立する。

【数49】

$$S_{55} = \delta_{55} \vee L_{55} \vee C_{56}$$

【数50】

$$S_{55} = \delta_{55} \vee C_{56}$$

【0124】式51が成立するので、式52が成立する。

【数51】

$$\delta_{55} = a_{55} \vee b_{55} \vee \kappa_{56}$$

【数52】

$$S_{55} = a_{55} \vee b_{55} \vee \kappa_{56} \vee C_{56}$$

【0125】同様に、式53が成立するが、式54が成立するので、式55が成立する。

【数53】

$$S_{55}^\dagger = \varepsilon_{55} \vee b_{55} \vee \lambda_{55}$$

【数54】

$$\varepsilon_{55} = a_{55} \vee L_{55} \vee \gamma_{56} = a_{55} \vee \gamma_{56}$$

【数55】

$$S_{55}^\dagger = a_{55} \vee \gamma_{56} \vee b_{55} \vee \lambda_{55} = a_{55} \vee b_{55} \vee \gamma_{56} \vee \lambda_{55}$$

【0126】式57が成立するので、式57と58が成立する。

【数56】

$$S_{55} = S_{55}^\dagger$$

【数57】

$$a_{55} \vee b_{55} \vee \kappa_{56} \vee C_{56} = a_{55} \vee b_{55} \vee \gamma_{56} \vee \lambda_{55}$$

【数58】

$$\kappa_{56} \vee C_{56} = \gamma_{56} \vee \lambda_{55}$$

39

【0127】ここで5つのMVC型オーバーラップ・ケースのそれぞれについて考察しなければならない。

【0128】定理5 オーバーラップ・ケース1及び5
ケース1及び5では、 $E(56:63) \geq B(56:63)$ である。定理1により $\lambda_{56}=1$ である。さらに、 $A(56:63) > B(56:63)$ なので、定理1により $\kappa_{56}=1$ である。最後に、 $A(56:63) + L(56:63)$ はアドレス空間の次の順次256バイト・ブロックに入る所以、キャリー γ_{56} も1である。これを数式5.9に代入すると、数式6.0が得られる。

【数5.9】

$$\kappa_{56} \wedge c_{56} = \gamma_{56} \wedge \lambda_{56}$$

【数6.0】

$$1 \wedge c_{56} = 1 \wedge 1$$

$$\overline{c_{56}} = 0$$

$$c_{56} = 1$$

【0129】定理5 オーバーラップ・ケース2及び4
ケース2及び4では、 $A(56:63) < B(56:63)$ かつ $E(56:63) < B(56:63)$ である。定理2により $\kappa_{56}=0$ かつ $\lambda_{56}=0$ である。ケース1の場合と同様に $A(56:63) + L(56:63)$ は256バイトの境界を横切るので、 $\gamma_{56}=1$ である。これらの恒等式から、数式6.1が成立する。

【数6.1】

$$0 \wedge c_{56} = 1 \wedge 0$$

$$c_{56} = 1$$

【0130】定理5 オーバーラップ・ケース3 ケース3では、 $A(56:63) < B(56:63)$ かつ $E(56:63) \geq B(56:63)$ である。定理2及び1から $\kappa_{56}=0$ かつ $\lambda_{56}=1$ である。 $A(56:63) + L(56:63)$ は256バイトの境界を横切らないので、 $\gamma_{56}=0$ である。これらの恒等式から、数式6.2が成立する。

【数6.2】

$$0 \wedge c_{56} = 0 \wedge 1$$

$$c_{56} = 1$$

【0131】only if の証明 only if を証明するには、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ であり、かつオペランドがMVC型オーバーラップを含まないとき、 c_{56} が0であることを示さなければならない。定理4から、考察しなければならないケースは、非オーバーラップ・ケース3と $\kappa_{56}=1$ のときのケース1、6、11である。

10

40

【0132】定理5 非オーバーラップ・ケース3 このケースでは、 $A(56:63) < B(56:63)$ かつ $E(56:63) < B(56:63)$ である。定理2から $\kappa_{56}=0$ かつ $\lambda_{56}=0$ である。 $A(56:63) + L(56:63)$ は256バイトの境界を横切らないので、 $\gamma_{56}=0$ である。したがって数式6.3が成立する。

【数6.3】

$$0 \wedge c_{56} = 0$$

$$c_{56} = 0$$

10

【0133】定理5 非オーバーラップ・ケース1及び6 これらのケースでは、 $\kappa_{56}=1$ の場合、 $0 \leq i \leq 55$ である。すべての i について $\delta_i=1$ である。したがって、 $\kappa_{56}=1$ である場合だけを考えればよい。さらに、 $E(56:63) < B(56:63)$ であるので、定理2から $\lambda_{56}=0$ である。最後に、 $A(56:63) + L(56:63)$ は256バイトの境界を横切るので、 $\gamma_{56}=1$ である。これらのキャリーを上記で導いたキャリー関係式に代入すると、数式6.4が得られる。

【数6.4】

$$1 \wedge c_{56} = 1 \wedge 0$$

$$\overline{c_{56}} = 1$$

20

$$c_{56} = 0$$

30

【0134】定理5 非オーバーラップ・ケース11 このケースでは、 $\kappa_{56}=1$ の場合、 $0 \leq i \leq 55$ である。すべての i について $\delta_i=1$ である。定理1から、 $\kappa_{56}=1$ の場合、 $A(56:63) \geq B(56:63)$ である。しかし、 $E(56:63) \geq A(56:63)$ であるので、 $E(56:63) \geq B(56:63)$ である。この結果から、定理1により $\lambda_{56}=1$ である。また $A(56:63) + L(56:63)$ は256バイトの境界を横切らないので、 $\gamma_{56}=0$ である。これらのキャリーを上記で導いたキャリー関係式に代入すると数式6.5が得られる。証明終り。

【数6.5】

$$1 \wedge c_{56} = 0 \wedge 1$$

$$\overline{c_{56}} = 1$$

50

$$c_{56} = 0$$

【0135】定理6 SAR2がアドレス空間の高位256バイト・ブロックにあり、SAR1が低位256バイト・ブロックにあるとき、SAR1がSAR2より上にマップされ、演算(SAR2-SAR1)+Lの結果、ビット位置56からビット位置55へのキャリーが

41

生じ、 $0 \leq i \leq 55$ であるすべてのビット位置 i で結果がすべて0となるならば、記憶オペランド1及び2はMVC型オーバーラップを有する。

【 0136】証明

i_f の証明

定理3により、2つのオペランドがMVC型オーバーラップを含み、SAR1が最低位256バイト・ブロックにありSAR2が最高位256バイト・ブロックにある場合にSAR1がSAR2のすぐ次の256バイト・ブロックにマップされるならば、 $SAR2 - SAR1 = 0 \leq i \leq 55$ である各ビット位置 i すべて1を生成する。LはOL(0:7)とその左に連結された56個の0からなり、定理5から $c_{56}=1$ なので、 $0 \leq i \leq 55$ である各ビット i はそのビットへの1のキャリーに加算された1個の1と0からなる。その結果は0であり左隣りのビットへのキャリーが生成され、したがって $0 \leq i \leq 55$ であるすべての i について結果は0となる。

【 0137】only i_f の証明 記憶オペランド1及び2がMVC型オーバーラップを有しないと仮定する。さらに、 $0 \leq i \leq 55$ であるすべての i について $S_i = 0$ であり、かつ $c_{56}=1$ であると仮定する。しかし、 $0 \leq i \leq 55$ であるすべての i について $S_i = 0$ であるなら、 $c_{56}=1$ のとき、 $0 \leq i \leq 55$ であるすべての i について $S_i = 1$ である。しかし、記憶オペランドがMVC型オーバーラップを有するならば、定理5から $0 \leq i \leq 55$ であるすべての i について $S_i = 1$ であり、 $c_{56}=1$ である。したがって、記憶オペランドがMVC型オーバーラップを有しないという仮定は誤っている。証明終り。

【 0138】 定理6は、MVC型オーバーラップの検出に使用される条件を提供する。要約すると、MVC型オーバーラップでの検出は次のようにして行う。まず、SAR2が仮想アドレス空間の最高位256バイト・ブロックにある(SAR2 HI)とき、SAR1が最低位256バイト・ブロックにある(SAR1 LO)ことを検出しなければならない。このケースが生じる(REMAP1)場合、SAR1を最高位256バイト・ブロックのすぐ上の256バイト・ブロックにマップしなければならない。そうでない場合は、SAR1は再マップされない。このSAR1のマッピングでは、SAR1をSA *40

$$(SAR2 - SAR1) + L = (B_2 + D_2 - B_1 - D_1) + L$$

【 0142】 この議論では、SAR2とSAR1のどちらの計算も折り返さないものと仮定する。言い換れば、24ビット・アドレス指定ではビット40から39へのキャリーが生じず、31ビット・アドレス指定ではビット33から32へのキャリーが生じず、64ビット・アドレス指定では先に論じたように折返しが発生し得ない。この折返しがないと仮定により、当面の議論で

42

*R2から減算する。その結果に、56個の0とオペランド長の連結であるLを加算する。この加算の結果が $0 \leq i \leq 55$ であるすべての i についてすべて0を含み、ビット56から55へのキャリーが1であるならば、オペランドはMVC型オーバーラップを含む。次に、これらの条件をさらに展開して、MVC型オーバーラップのハードウェア予測に適した形にすることを追求する。

【 0139】 MVC型オーバーラップを示す可能性のあるESA/390命令のアドレス計算は、基底に12ビットの変位を加算することからなる。24ビットのアドレス指定を用いるアクセス・レジスタ・モードでは、この加算は、24ビットの基底に8個のゼロを連結したものと連結したアクセス・レジスタに12ビットの変位を加算することからなる。31ビット・アドレス指定を用いるアクセス・レジスタ・モードでは、この加算は、31ビットの基底に1個のゼロを連結したものと連結したアクセス・レジスタに12ビットの変位を加算することからなる。64ビット・モードでは、その計算は、63ビットの基底に連結した1個の0に12ビットの変位を加算することからなる。以下では、12ビットの変位DP(0:11)の左に52個の0を連結した64ビット数を D_2 と D_1 で表すこととする。この形の記憶オペランド2に対する変位を D_2 で表し、記憶オペランド1に対する変位を D_1 で表す。 D_2 及び D_1 の下位12ビットをそれぞれ d_{2i} 及び d_{1i} で表す。ただし、 $52 \leq i \leq 63$ である。さらに、オペランド1及び2の基底をそれぞれ B_1 及び B_2 で表し、個々のビットを b_{1i} 及び b_{2i} で表す。ただし、 $0 \leq i \leq 63$ である。この形で表した基底は、上記のアクセス・レジスタ・モードの実際の基底とアクセス・レジスタの連結を含むことを理解されたい。前にSAR1を表すのに使用したBがここではアドレス計算の基底を表すことに留意されたい。

【 0140】 この表記法を使うと、SAR2とSAR1は数式66のように計算される。

$$SAR2 = B_2 + D_2$$

$$SAR1 = B_1 + D_1$$

【 0141】 そうすると、 $(SAR2 - SAR1) + L$ は数式67のようになる。

【 数67】

これらのビット位置へのキャリーを抑制するという複雑さを回避することができる。この複雑さについては後で考察する。

【 0143】 2の補数の算術を使用すると、 $(SAR2 - SAR1) + L$ の計算を数式68で表すことができる。

【 数68】

$$(SAR2 - SAR1) + L = (B_2 + D_2 + B_1 + D_1 + 1 + 1) + L$$

43.

【0144】上式でバーB_i及びバーD_iはそれぞれB_i*9のような2-1加算に還元できる。
及びD_iの補数を示す。括弧内の結果の計算は、数式6 * 【数69】

$$\begin{array}{cccccccccc} b_{20} & b_{21} & \dots & b_{251} & b_{252} & b_{253} & b_{254} & b_{255} & b_{256} & \dots & b_{262} & b_{263} \\ 0 & 0 & \dots & 0 & d_{252} & d_{253} & d_{254} & d_{255} & d_{256} & \dots & d_{262} & d_{263} \end{array}$$

$$\overline{b_{10} b_{11} \dots b_{151} b_{152} b_{153} b_{154} b_{155} b_{156} \dots b_{162} b_{163}}$$

$$\begin{array}{cccccccccc} e_0 & e_1 & \dots & e_{51} & e_{52} & e_{53} & e_{54} & e_{55} & e_{56} & \dots & e_{62} & e_{63} \\ f_1 & f_2 & \dots & f_{52} & f_{53} & f_{54} & f_{55} & f_{56} & f_{57} & \dots & f_{63} & 1 \end{array}$$

$$\overline{1 \ 1 \ \dots \ 1 \ d_{152} d_{153} d_{154} d_{155} d_{156} \dots d_{162} d_{163}}$$

$$\begin{array}{cccccccccc} g_0 & g_1 & \dots & g_{51} & g_{52} & g_{53} & g_{54} & g_{55} & g_{56} & \dots & g_{62} & g_{63} \\ h_1 & h_2 & \dots & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & \dots & h_{63} & 1 \end{array}$$

【0145】個別ビットg_iを有するGと、Hが左に1ビットずつシフトされ、シフトによる空所が「ホット1」で埋められた、個別ビットh_iを有するHの2-1加算の結果、B₂+D₂-B₁-D₂となる。

【0146】この時点で、これまで使用してきた表記法を修正して、SAR2-SAR1を求める際の上記の考察を含める必要がある。GとシフトされたHの2-1加算から、SAR2-SAR1が得られるが、これを前に個別ビットδ_iを有するΔで表したので、GとシフトされたHの2-1加算の結果をこの表記法で表すことにする。したがって、数式70が成立する。

【数70】

$$\begin{array}{cccccc} g_0 \dots g_{55} & g_{56} \dots g_{62} & g_{63} \\ h_1 \dots h_{56} & h_{57} \dots h_{63} & 1 \end{array}$$

$$\overline{\delta_0 \dots \delta_{55} \ \delta_{56} \dots \delta_{62} \ \delta_{63}}$$

【0147】同様に、κ_iは、この2-1加算ではビットiからi-1へのキャリーという意味になる。さらに、以前の表記法との整合性を保つため、ΔとLの2-1加算の結果をSで表し、その個別ビットをS_iで表することにする。したがって、数式71が成立する。

【数71】

$$\begin{array}{cccccc} \delta_0 \dots \delta_{55} & \delta_{56} \dots \delta_{62} & \delta_{63} \\ L_0 \dots L_{55} & L_{56} \dots L_{62} & L_{63} \end{array}$$

$$\overline{S_0 \dots S_{55} \ S_{56} \dots S_{62} \ S_{63}}$$

【0148】この2-1加算でのビット位置iからi-1へのキャリーをc_iで表すことにする。

【0149】SAR2-SAR1+Lの結果Sは、Gと

シフトされたHとLのCSA3-2加算として計算することもでき、個別ビットσ_iを有する和と個別ビットω_iを有するキャリーを生成する。個別ビットω_iは、加算の前に左へ1ビット位置シフトされたωと加算される。この2-1加算の結果はSAR2-SAR1+Lを生じるが、これを個別ビットΣ_iを有するΣで表して、2つの2-1加算器を使った計算と区別する。したがって、数式72のようになる。

【数72】

$$g_0 \dots g_{55} \ g_{56} \dots g_{62} \ g_{63}$$

$$h_1 \dots h_{56} \ h_{57} \dots h_{63} \ 1$$

$$L_0 \dots L_{55} \ L_{56} \dots L_{62} \ L_{63}$$

$$\sigma_0 \dots \sigma_{55} \ \sigma_{56} \dots \sigma_{62} \ \sigma_{63}$$

$$\omega_1 \dots \omega_{56} \ \omega_{57} \dots \omega_{63} \ 0$$

$$\Sigma_0 \dots \Sigma_{55} \ \Sigma_{56} \dots \Sigma_{62} \ \Sigma_{63}$$

【0150】δ_iとシフトされたω_iの2-1加算におけるiからi-1へのキャリーをφ_iで表すこととする。

【0151】定理6から、c₅₆=1かつ0≤i≤55であるすべてのiについてS_i=0である場合に、MVC型オーバーラップが存在する。0≤i≤55であるすべてのiについてS_i=0であるとの予測をS₌₀で表し、MVC型オーバーラップをO_{MVC}で表すと、数式73のようになる。

【数73】O_{MVC}=c₅₆S₌₀

【0152】O_{MVC}が1であるためには、c₅₆が1でなければならず、したがってc₅₆=1の場合についてのみS₌₀を決定すればよい。c₅₆=1のときのS₌₀を決定することを数式74で表すことにする。次の定理が、数式75を求める基礎となる。

45

【 数74 】

$$S \neq 0$$

【 数75 】

$$S \neq 0$$

【 0153 】 定理7 $c_{56}=1$ である場合、 $\kappa_{56}=1$ のときは $\phi_{56}=\omega_{56}=1$ 、 $\kappa_{56}=0$ のときは $\phi_{56}=1$ かつ $\omega_{56}=0$ または $\phi_{56}=0$ かつ $\omega_{56}=1$ である。

【 0154 】 証明 結合法則により、3つの数の加算は加算の順序にかかわらず同じ結果を生じるので、数式7 6 が成立する。

【 数76 】 $S_{55} = \Sigma_{55}$

【 0155 】 しかし、数式7 7 と 7 8 が成立し、したがって数式7 9 が成立する。

【 数77 】

$$S_{55} = c_{56} \vee \delta_{55} \vee L_{55} = c_{56} \vee \delta_{55} \vee 0 = c_{56} \vee \delta_{55}$$

【 数78 】

$$\Sigma_{55} = \omega_{56} \vee \sigma_{55} \vee \phi_{56}$$

【 数79 】

$$c_{56} \vee \delta_{55} = \omega_{56} \vee \sigma_{55} \vee \phi_{56}$$

【 0156 】 数式8 0 と 8 1 から数式8 2 が得られる。

【 数80 】

$$\delta_{55} = g_{55} \vee h_{56} \vee \kappa_{56}$$

【 数81 】

$$\sigma_{55} = g_{55} \vee h_{56} \vee L_{55} = g_{55} \vee h_{56} \vee 0 = g_{55} \vee h_{56}$$

【 数82 】

$$c_{56} \vee g_{55} \vee h_{56} \vee \kappa_{56} = \omega_{56} \vee g_{55} \vee h_{56} \vee \phi_{56}$$

$$c_{56} \vee \kappa_{56} = \omega_{56} \vee \phi_{56}$$

【 0157 】 $c_{56}=1$ である場合、数式8 3 が成立する。

【 数83 】

$$\overline{\kappa_{56}} = \omega_{56} \vee \phi_{56}$$

【 0158 】 $\kappa_{56}=0$ である場合、数式8 4 が成立する。

【 数84 】

$$1 = \omega_{56} \vee \phi_{56}$$

【 0159 】 この恒等式が満たされるのは、 $\omega_{56}=1$ かつ $\phi_{56}=0$ または $\omega_{56}=0$ かつ $\phi_{56}=1$ の場合である。したがって、 $\kappa_{56}=0$ のとき、 $\omega_{56}=1$ かつ $\phi_{56}=0$ または $\omega_{56}=0$ かつ $\phi_{56}=1$ である。

【 0160 】 $\kappa_{56}=1$ である場合、数式8 5 が成立する。

【 数85 】

46

$$0 = \omega_{56} \vee \phi_{56}$$

【 0161 】 ω_{56} と ϕ_{56} がこの式を満足するには、 $\omega_{56} = \phi_{56} = 1$ または $\omega_{56} = \phi_{56} = 0$ である。後は、 $\omega_{56} = \phi_{56} = 1$ であることを示せばよい。 κ_{56} の式は数式8 6 で現れる。

【 数86 】 $\kappa_{56} = g_{56} h_{57} + g_{56} \kappa_{57} + h_{57} \kappa_{57}$

【 0162 】 まず、 $\kappa_{57}=0$ であると仮定する。この場合、数式8 7 が成立し、したがって $\kappa_{56}=1$ である場合、 $g_{56}=1$ かつ $h_{57}=1$ である。したがって、数式8 8 が成立するので $\omega_{56}=1$ である。

【 数87 】 $\kappa_{56} = g_{56} h_{57}$ 【 数88 】 $\omega_{56} = g_{56} h_{57} + L_{56} h_{57} + L_{56} g_{56}$

【 0163 】 キャリ一関係式に $\omega_{56}=1$ 、 $c_{56}=1$ 、 $\kappa_{56}=1$ を代入すると、数式8 9 が得られる。

【 数89 】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【 0164 】 したがって、 $\kappa_{57}=0$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ のとき、 $\omega_{56}=1$ かつ $\phi_{56}=1$ である。

【 0165 】 次に $\kappa_{57}=1$ であると仮定すると、数式9 0 が成立する。

【 数90 】 $\kappa_{56} = g_{56} + h_{57}$

【 0166 】 したがって、 $\kappa_{56}=1$ である場合、 $g_{56}=1$ または $h_{57}=1$ である。さらに、数式9 1 が成立する。

【 数91 】 $c_{56} = \delta_{56} L_{56} + \delta_{56} c_{57} + L_{56} c_{57}$

【 0167 】 $c_{57}=1$ であると仮定すると、数式9 2 が成立し、 $c_{56}=1$ なので $\delta_{56}=1$ または $L_{56}=1$ である。まず $L_{56}=1$ であると仮定すると、数式9 3 が成立し、 $g_{56}=1$ または $h_{57}=1$ なので、数式9 4 が得られる。

【 数92 】 $c_{56} = \delta_{56} + L_{56}$ 【 数93 】 $\omega_{56} = g_{56} + h_{57}$ 【 数94 】 $\omega_{56} = 1$

【 0168 】 キャリ一関係式に代入すると、数式9 5 が得られる。

【 数95 】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【 0169 】 今度は $\delta_{56}=1$ であると仮定すると、数式9 6 が成立し、かつ $\kappa_{57}=1$ であると仮定したので、数式9 7 が成立する。

50

47

【数96】

$$\delta_{56} = g_{56} \vee h_{57} \vee \kappa_{57}$$

【数97】

$$\delta_{56} = \overline{g_{56} \vee h_{57}}$$

$$1 = \overline{g_{56} \vee h_{57}}$$

【0170】しかし、 $g_{56}=1$ または $h_{57}=1$ ので、
 $g_{56}=1$ かつ $h_{57}=1$ となる。したがって、数式9.8 が
 成立するので、 $\omega_{56}=1$ となる。キャリー関係式に代入
 すると数式9.9 が得られる。

【数98】 $\omega_{56} = g_{56}h_{57} + g_{56}L_{56} + h_{57}L_{56}$

【数99】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0171】したがって、 $\kappa_{57}=1$ 、 $c_{57}=1$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ の場合、 $\omega_{56}=1$ かつ $\phi_{56}=1$ である。

【0172】最後に $c_{57}=0$ であると仮定すると、数式1.0.0 が成立し、したがって $c_{56}=1$ である場合、 $\delta_{56}=1$ かつ $L_{56}=1$ である。数式1.0.1 が成立するので、数式1.0.2 が得られる。

【数1.0.0】 $c_{56} = \delta_{56}L_{56} + \delta_{56}c_{57} + L_{56}c_{57}$
 $c_{56} = \delta_{56}L_{56}$

【数1.0.1】 $\omega_{56} = g_{56}h_{57} + g_{56}L_{56} + h_{57}L_{56}$ 【数1.0.2】 $\omega_{56} = g_{56} + h_{57}$

【0173】しかし、仮定したように $\kappa_{57}=1$ の場合は、 $g_{56}=1$ または $h_{57}=1$ であり、したがって $\omega_{56}=1$ となる。再度キャリー関係式に代入すると、数式1.0.3 が得られる。

【数1.0.3】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0174】したがって、 $\kappa_{57}=1$ 、 $c_{57}=0$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ の場合、 $\omega_{56}=1$ かつ $\phi_{56}=1$ であり、したがって $c_{56}=1$ かつ $\kappa_{56}=1$ である場合、 $\omega_{56}=\phi_{56}=1$ となる。証明終り。

【0175】次に $\Sigma(0:55)$ の計算について考える。 Σ は σ と左に1ビット位置シフトされた ω の2-1 加算によって算出される。これは数式1.0.4 で表される。

50

48

【数1.0.4】

$$\sigma_0 \sigma_1 \dots \sigma_{54} \sigma_{55}$$

$$\omega_1 \omega_2 \dots \omega_{55} \omega_{56}$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0176】 $c_{56}=1$ のとき、 $0 \leq i \leq 5$ である i についてすべての Σ_i が0 であることを決定しなければならない。この決定を行うには、定理7 を用いて Σ の計算の表現を修正する。定理7 から、 $c_{56}=1$ のときは、 $\kappa_{56}=1$ のとき ϕ_{56} と ω_{56} が共に1 でなければならず、 $\kappa_{56}=0$ のとき ϕ_{56} と ω_{56} のうち一方だけが1 となり得る。したがって、この加算は数式1.0.5 のような3-1 加算として表すことができる。

【数1.0.5】

$$\sigma_0 \sigma_1 \dots \sigma_{54} \sigma_{55}$$

$$\omega_1 \omega_2 \dots \omega_{55} \omega_{56}$$

$$0 \ 0 \dots 0 \ 1$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0177】加算をこの形で表すことの利点は、下記の諸定理から出てくる。提示を簡単にするため、 $0 \leq i \leq 5$ であるすべての i について σ_i を P_{xi} で表し、 κ_{56} と連結された ω_i ($1 \leq i \leq 5$) を P_{yi} ($1 \leq i \leq 5$) で表すことになる。そうすると上記の式は数式1.0.6 のようになる。

【数1.0.6】

$$P_{x0} P_{x1} \dots P_{x54} P_{x55}$$

$$P_{y1} P_{y2} \dots P_{y55} P_{y56}$$

$$0 \ 0 \dots 0 \ 1$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0178】定理8 $\Sigma_i=0$ である場合、 P_x と P_y との加算によって生成される位置 i でのキャリー ν_i は、 $0 \leq i \leq 5$ であるすべての i について1 に等しい。

【0179】証明 定理8 は帰納法で証明できる。

【0180】帰納の基底 ビット位置55における和の式は数式1.0.7 のようになる。

【数1.0.7】

$$\Sigma_{55} = P_{x55} \vee P_{y55} \vee 1$$

$$= \overline{P_{x55} \vee P_{y55}}$$

【0181】 $\Sigma_{55}=0$ であると仮定すると数式1.0.8 が成立し、したがって数式1.0.9 が成立する。

【数1.0.8】

$$\overline{P_{x55} \vee P_{y55}} = 0,$$

49

【数109】

$$P_{x_{55}} \vee P_{y_{55}} = 1.$$

【0182】ビット位置55で生成されるキャリーカー v_{55} は数式110で表される。

【数110】

$$v_{55} = P_{x_{55}} P_{y_{56}} + P_{x_{55}} (1) + P_{y_{56}} (1)$$

【0183】上式で+は論理ORを表す。この式は数式111に還元される。

$$【数111】 v_{55} = P_{x_{55}} + P_{y_{56}}$$

【0184】 $\Sigma_{55}=0$ の場合、数式112が成立し、かつP_{x₅₅}とP_{y₅₆}のどちらか一方が1であるが、両方が1ではない場合は排他的ORは1となるので、数式113が成立し、したがって数式114が成立する。
10

【数112】

$$P_{x_{55}} \vee P_{y_{56}} = 1,$$

$$【数113】 P_{x_{55}} + P_{y_{56}} = 1$$

$$【数114】 v_{55} = 1$$

【0185】したがって、ビット55での加算の和 Σ_{55} が0に等しい場合、ビット55からビット54へのキャリーカーは1であることが保証される。
20

【0186】帰納のステップ $\Sigma_i=0$ かつ $v_{i+1}=1$ であると仮定する。位置iでの和 Σ_i は数式115で表されるので、数式116または117が成立する。

【数115】

$$\Sigma_i = P_{x_i} \vee P_{y_{i+1}}, \forall v_{i+1},$$

【数116】

$$P_{x_i} \vee P_{y_{i+1}}, \forall 1 = 0,$$

【数117】

$$\overline{P_{x_i} \vee P_{y_{i+1}}} = 0$$

30

$$P_{x_i} \vee P_{y_{i+1}} = 1.$$

【0187】位置iでのキャリーカーは数式118で表され*

$$v_i = P_{x_1} P_{y_{i+1}} + P_{x_1} v_{i+1} + P_{y_{i+1}} v_{i+1}$$

$$\begin{aligned} &= P_{x_1} P_{y_{i+1}} + P_{x_1} v_{i+1} (P_{y_{i+1}} + \overline{P_{y_{i+1}}}) P_{y_{i+1}} v_{i+1} (P_{x_1} + \overline{P_{x_1}}) \\ &= P_{x_1} P_{y_{i+1}} + P_{x_1} \overline{P_{y_{i+1}}} v_{i+1} + P_{y_{i+1}} \overline{P_{x_1}} v_{i+1} \\ &= P_{x_1} P_{y_{i+1}} + (P_{x_1} \vee P_{y_{i+1}}) v_{i+1} \end{aligned}$$

【0194】仮定により、数式126が成立するので、数式127が成立する。

【数126】

$$P_{x_1} \vee P_{y_{i+1}} = 1,$$

$$【数127】 v_i = v_{i+1}$$

【0195】上式がすべてのiについて成立しなければならず、かつキャリーカーインが1なので、数式128が成立する。

$$【数128】 v_i = v_{i+1} = \dots = 1$$

【0196】 $\Sigma_i=1$ とする。ただし Σ_i は数式129で
50

*る。

【数118】

$$v_i = P_{x_i} P_{y_{i+1}} + P_{x_i} v_{i+1} + P_{y_{i+1}} v_{i+1}$$

【0188】 $v_{i+1}=1$ なので、数式119が成立する。

$$【数119】 v_i = P_{x_i} P_{y_{i+1}} + P_{x_i} + P_{y_{i+1}}$$

$$= P_{x_i} + P_{y_{i+1}}$$

【0189】しかし、上記から数式120が成立し、したがって数式121及び数式122が成立する。

【数120】

$$P_x \vee P_{y_{i+1}} = 1,$$

$$【数121】 P_{x_i} + P_{y_{i+1}} = 1$$

$$【数122】 v_i = 1$$

【0190】上記のステップ1と2から、数学的帰納により $0 \leq i \leq 55$ であるiについて $v_i=1$ となる。証明終り。

【0191】定理9 $0 \leq i \leq 55$ であるiについて数式123が成立するならば和 Σ は0である。

【数123】

$$P_x \vee P_{y_{i+1}} = 1$$

【0192】証明

ifの証明

$\Sigma=0$ ならば、任意のiについて定理8から $v_{i+1}=1$ であり、したがって $\Sigma_i=0$ なので数式124が成立する。

【数124】

$$P_x \vee P_{y_{i+1}} = 1.$$

$$【0193】 only if の証明$$

v_i で生成されるキャリーカーについて考える。このキャリーカーは数式125で表される。

【数125】

表される。

【数129】

$$\Sigma_i = P_{x_i} \vee P_{y_{i+1}}, \forall v_{i+1}.$$

【0197】ただし、数式130が成立し、したがって数式131が成立する。

【数130】

$$P_x \vee P_{y_{i+1}} = 1$$

$$【数131】 v_{i+1} = 0$$

【0198】したがって、 $\Sigma_i=1$ であると仮定すると

51

矛盾が生じ、したがってこの仮定は誤っている。したがって Σ_i は0でなければならない。証明終り。

【 0 1 9 9 】 数式1 3 2 が0となるための条件は、定理8と9から導かれる。この条件とは、 $0 \leq i \leq 5 5$ であるすべての i について数式1 3 3 が成立することである。 P_{xi} と P_{yi+1} に適当な値を代入すると数式1 3 4 が得られる。

【 数1 3 2 】

S₀[†]

【 数1 3 3 】

$$P_{xi} \vee P_{yi+1} = 1$$

10

【 数1 3 4 】

$$S_{=0}^{\dagger} = (\sigma_0 \vee \omega_1) (\sigma_1 \vee \omega_2) \dots (\sigma_{54} \vee \omega_{55}) (\sigma_{55} \vee K_{56})$$

【 0 2 0 0 】 数式1 3 5 が決定されると、既に示したように、数式1 3 6 から O_{MVC} を求めることができる。

【 数1 3 5 】

S₀[†]

【 数1 3 6 】

$$O_{MVC} = c_{56} S_{=0}^{\dagger}$$

20

【 0 2 0 1 】 MVC 型オーバーラップを予測するための制限のある装置が開発されたばかりである。この装置は、第1及び第2のオペランド・アドレスの計算中にラップが起こらないと仮定されているので、一般的ではない。さらに、第1オペランド・アドレスが有効アドレス空間の最低位2 5 6 バイト・ブロックを指すとき、第2オペランド・アドレスが最高位2 5 6 バイト・ブロックを指さないと仮定されている。したがって、第1オペランド・アドレスを第2オペランド・アドレスより上に再マップする必要はなかった。一般的の場合に MVC 型オーバーラップを予測するには、こうした状況を考慮して、この以前の予測方式をオーバーラップが正しく予測できるように補正しなければならない。

【 0 2 0 2 】 オーバーラップが正しく予測できるようにするために行わなければならない補正を表3に示す。この表に列挙した補正是3つのアドレス指定モードのすべてに適用される。表中で REMAP1 は、SAR2 がアドレス空間の最高位2 5 6 バイト・ブロックにあり、SAR1 が最低位2 5 6 バイト・ブロックにあるときに発生する。これが発生するときは、SAR2 が存在する2 5 6 バイト・ブロックのすぐ上の2 5 6 バイト・ブロックに SAR1 を再マップしなければならない。これは、アドレス空間を指定するビットの左MSB に隣接するビット位置を強制的に1 にすることによって実施できる。このビット位置は、24ビット・アドレス指定の場合はビット3 9、31ビット・アドレス指定の場合はビット3 2 である。64ビット・アドレス指定の場合は、前述の理由から再マップは不要である。SAR1 及び SAR

30

40

50

52

2のアドレス計算中に折返しが発生したことを示すのにそれぞれWRAP1 及びWRAP2 を使用する。その際に基底が変位に加算される。WRAP1 及びWRAP2 の条件は、基底が変位と加算されるとき、アドレス空間を指定するビットのMSB からキャリーが生じることを検出することによって識別できる。したがって、24ビット及び31ビットの場合に折返しが発生したことを判定するために、それぞれビット4 0 及び3 3 からのキャリーを検出しなければならない。64ビット・アドレス指定での $2^{64}-1$ のアドレス空間及び64ビット・アドレスのMSB は0でなければならないので、64ビット・アドレス指定モードでは折返しは生じ得ない。しかし24ビットまたは31ビット・アドレス指定モードで折返しが発生するときは、オーバーラップが正しく予測できるように、アドレス空間のMSB から生じるキャリーを抑制しなければならない。

【 0 2 0 3 】 上記の議論から、表3に列挙した補正を適用しなければならないのは、24ビット及び31ビット・アドレス指定モードのときだけである。この2つのモードで必要な補正の違いは、キャリーを抑制または強制すべきビット位置だけである。このため、以下の議論では24ビット・アドレス指定の場合の補正の適用のみについて述べる。31ビット・アドレス指定の場合の結果は、展開なしに提示されるであろう。前述のように、24ビット・アドレス指定で強制または抑制しなければならないキャリーは、BとDの加算中に生成されるビット位置4 0 からビット位置3 9 へのキャリーである。このキャリーは、加算B+Dにその値を加算することによって強制できる。このキャリーの値は数式1 3 7 で与えられ、ビット位置3 9 に1が生じる。

【 数1 3 7 】 0 0 ... 0 1 0 ... 0 0

【 0 2 0 4 】 同様に、キャリー1値の2の補数を加えることによってキャリーが抑制できる。上式のキャリーの値を使うと、2の補数は数式1 3 8 のようになる。

【 数1 3 8 】

1 1 ... 1 0 1 ... 1 1

0 0 ... 0 0 0 ... 0 1

1 1 ... 1 1 0 ... 0 0

【 0 2 0 5 】 したがって、B+Dに(1 1 ... 1 1 0 ... 0 0)を加えることによって、キャリーが抑制できる。しかし、 MVC 型オーバーラップを予測するには、数式1 3 9 の算術演算から $0 \leq i \leq 5 5$ であるビット位置についてすべて0を検出しなければならない。

【 数1 3 9 】 $B_2 + D_2 - (B_1 + D_1) + L$

【 0 2 0 6 】 第1オペランドのアドレス計算によるキャリーを強制するには、キャリーの値を $B_1 + D_1$ に加算しなければならない。しかし、上式では $B_1 + D_1$ が減算さ

れるので、第1オペラント・アドレスの計算のためにキャリーを強制するには、キャリー一値の2の補数をMVC型オーバーラップでの計算に加算しなければならない。同様に、第1オペラントのアドレス計算からのキャリーを抑制するには、上式にキャリー一値を加算しなければならない。

【 0 2 0 7 】この議論から、表3のケース1ないし4で適用すべき補正は明らかになるであろう。第1のケースでは補正是不要である。これは、 $B_2 + D_2$ 及び $B_1 + D_1$ の結果に0を加算することによって実施できる。その正味の効果は、 $B_2 + D_2 - (B_1 + D_1)$ に0を加算することである。第1オペラントの計算中に折返しが発生する第2のケースでは、 $B_1 + D_1$ の間に発生するキャリーを抑制しなければならない。したがって、このキャリーを抑制するには、上記の理由から、 $B_2 + D_2 - (B_1 + D_1)$ にキャリーの値を加算しなければならない。ケース3では第2オペラントのアドレス計算中に折返しが発生する。この場合、このアドレス計算によるキャリーは、 $B_2 + D_2 - (B_1 + D_1)$ にキャリー一値の2の補数を加算することによって抑制される。最後に第4のケースでは、両方のオペラントのアドレス計算による折返しを抑制しなければならない。しかし、これらのキャリーを抑制するために加算しなければならない値は加法逆数であり、したがって正味の結果として、この補正のために0の値を加算しなければならない。以上の結果を表4の最初の4行にまとめてある。

【 0 2 0 8 】表3の残り4つのケースでは、アドレス空間の最高位256バイト・ブロックのすぐ上にSAR1を再マップする必要がある。前述のように、これは、SAR1を生成するためのアドレス計算の結果の適当なビット位置に強制的に1個の1を入れることによって実施される。これは、そのビット位置にキャリーを強制するのと同じことである。したがって、どちらかのオペラント・アドレスの計算中に折返しが起こらないのに再マップが必要なケース5の場合は、キャリーの値を $B_1 + D_1$ に加算しなければならず、したがって $B_2 + D_2 - (B_1 + D_1)$ にキャリー一値の2の補数を加算しなければならない。ケース6の場合は、第1オペラントの計算で折返しが発生するのに、再マップが必要である。すなわち、 $B_1 + D_1$ がキャリーを発生し、それによってSAR1がアドレス空間の最高位256バイト・ブロックのすぐ上に自動的に再マップされる。 $B_2 + D_2 - (B_1 + D_1)$ に0を加算することによって実施できる補正は必要でない。24ビットの基底に12ビットの変位を加算しても、折返しは発生し得ず、24ビット・アドレス空間の最高位256バイト・ブロックにあるアドレスが発生し

得ないので、ケース7及び8は発生し得ない。したがって、これらのケースの対応する項目は表4に含まれていない。31ビット・アドレス指定での同じ分析の結果を表5に示す。表4及び5からわかるように、すべてのアドレス指定モードのすべてのケースの考察から、独特な5つの補正が生じる。これら独特な補正を表6にまとめ示す。これら独特な補正と表4及び5の各ケースの間のマップを表7に示す。

【 0 2 0 9 】定理3ないし6に含まれる仮定と矛盾しないSAR2-SAR1の値を生成するには、 $B_2 + D_2 - (B_1 + D_1)$ の計算に表6の補正を適用しなければならない。したがって、 $B_2 + D_2 - (B_1 + D_1)$ にオペラント記憶域長を加算する前に、補正值を適用しなければならない。これは、補正值を加算し、加算の結果がSAR2-SAR1となる2つのエンティティを得るために3-2CSA段がもう1つ必要なことを示唆する。これらの定理で要求されるように折返しが除去され、再マップが強制される。次にCSAの出力を上記のG及びシフトされたHと同様に扱ってオーバーラップを検出することができる。

【 0 2 1 0 】しかし、次のようにしてさらに高い並列性を得ることができる。まず、3-2加算器を使って $B_2 + バーB_1 + バーD_1$ を和とキャリーに還元することができる。補正が行われない場合は、第2の3-2CSAによってこの和とキャリーを D_2 と加算し、前述のように使用されるG及びシフトされたHを生成することができる。しかし、補正が必要な場合は、補正值を D_2 と連結することによって加算が実施できるので、 D_2 と補正值の2-1加算が直ちに生成できる。たとえば、表6の補正3で、補正值と D_2 の2-1加算は数式140のようになる。

【 数140 】

$$0\ 0\ :\ldots\ 0\ 0\ 0\ \ldots\ d_{252}\ d_{253}\ \ldots\ d_{263}$$

$$1\ 1\ :\ldots\ 1\ 1\ 0\ \ldots\ 0\ 0\ \ldots\ 0$$

$$1\ 1\ :\ldots\ 1\ 1\ 0\ \ldots\ d_{252}\ d_{253}\ \ldots\ d_{263}$$

【 0 2 1 1 】次にこの値を $B_2 + バーB_1 + バーD_1$ の3-2加算からの和及びキャリーに加算すると、補正されたGとHを生成することができ、これは制限つきのケースでのオーバーラップ検出の場合と同様に扱うことができる。したがって数式141が得られる。

【 数141 】

55
 $b_{20} b_{21} \dots b_{251} b_{252} b_{253} b_{254} b_{255} b_{256} \dots b_{262} b_{263}$

$\overline{b_{10}} \overline{b_{11}} \dots \overline{b_{151}} \overline{b_{152}} \overline{b_{153}} \overline{b_{154}} \overline{b_{155}} \overline{b_{156}} \dots \overline{b_{162}} \overline{b_{163}}$

1 1 ... 1 $\overline{d_{152}} \overline{d_{153}} \overline{d_{154}} \overline{d_{155}} \overline{d_{156}} \dots \overline{d_{162}} \overline{d_{163}}$

$e_0 e_1 \dots e_{51} e_{52} e_{53} e_{54} e_{55} e_{56} \dots e_{62} e_{63}$

$f_1 f_2 \dots f_{52} f_{53} f_{54} f_{55} f_{56} f_{57} \dots f_{63} 1$

1 1 ... 0 $d_{252} d_{253} d_{254} d_{255} d_{256} \dots d_{262} d_{263}$

$g_0 g_1 \dots g_{51} g_{52} g_{53} g_{54} g_{55} g_{56} \dots g_{62} g_{63}$

$h_1 h_2 \dots h_{52} h_{53} h_{54} h_{55} h_{56} h_{57} \dots h_{63} 1$

【 0 2 1 2 】 最高の並列性を引き出すには、2つの3-2 C S Aのシーケンスによって $B_2 - B_1 - D_1 + D_2$ の計算に表6に示したすべての潜在的補正を適用して、すべての潜在的G及びHを生成することができる。これらを加算すると所望のS A R 2 - S A R 1 の値が得られる。
これらはそれぞれ前に示したように使用され、そのケ

スが決定された後に適切な標識が使用のために選ばれる。ケースの決定は、表7に従ってアドレス指定モード、WRAP1、WRAP2、REMAP1の適切な復号により、これらの計算と平行して行われる。

【 表3 】

M V C型オペランドのオーバーラップを予測するために有効アドレス空間のM S Vからのキャリーを強制または抑制するための要件

ケース	REMAP1	WRAP2	WRAP1	処置
ケース1	なし	なし	なし	補正処置なし
ケース2	なし	なし	あり	$B_1 + D_1$ に対するMSBからのキャリーを抑制する
ケース3	なし	あり	なし	$B_2 + D_2$ に対するMSBからのキャリーを抑制する
ケース4	なし	あり	あり	$B_2 + D_2$ に対するMSB及び $B_1 + D_1$ からのキャリーを抑制する
ケース5	あり	なし	なし	$B_1 + D_1$ に対するMSBからのキャリーを強制する
ケース6	あり	なし	あり	$B_1 + D_1$ に対するMSBからのキャリーを許容する
ケース7	あり	あり	なし	不能
ケース8	あり	あり	あり	不能

【表4】

【表5】

【表6】

【表7】

31ビット・アドレス指定モードでのMVC型オーバーラップの提出用の補正	
補 正	マッピング
COR1	すべてのアドレス指定モード ケース1, 4, 6
COR2	AR24ビットモード ケース2
COR3	AR24ビットモード ケース3及び5
COR4	AR31ビットモード ケース 2
COR5	AR31ビットモード ケース3及び5

【0213】好ましい実施例

ここで本発明の詳細な説明に移ると、図4には、MVC型の破壊的な記憶オペランド・オーバーラップを予測するのに使用されるハードウェアの概要が示してある。レジスタ1 h、r h、l a、r a はそれぞれ SAR2 及び SAR1 を算出するための基底及び変位の値である B2、D2、B1、D1 を格納する。これらのレジスタは、折返し条件 WRAP2 と WRAP1 を予測し、かつ図5 及び図6 に示すように SAR2 がアドレス空間の高位256 バイト・ブロックにあるかどうか (SAR2H1) または SAR1 がアドレス空間の低位256 バイト

40 ・ ブロックにあるかどうかを判定するためのブロックに出力を供給する。SAR2HI と SAR2LO は、図4に示すようにANDされて、再マップが必要なことを示すREMAPP1（図4には記載せず）を生成する。REMAPP1、WRAP1、WRAP2 は、表6に従ってアドレス指定モードと共に復号されて、オーバーラップを判定するためにどのオーバーラップ補正（以下の式では、OMVCCOR1、OMVCCOR2、OMVCCOR3で示す）を使用すべきかを示すCOR（1:5）を生成する（図4のREMAPP1、REMAPP2、WRAP1、WRAP2 復号ブロック）。COR（1:5）はオーバーラップ予測

50

59

ブロックに送られ、そこで適当なオーバーラップ標識の実際の選択が行われる。オーバーラップ予測ブロックは、前述の諸アルゴリズムを実施したものである。このブロックの詳細は、図5及び6に示す。

【 0 2 1 4 】図5及び6では、キャリー・セーブ加算器CSA1を使って、 $1 h + l a _N + r a _N (B_2 + B_1 + D_1)$ の3-2加算を実行する。CSA1によって生成される和EとキャリーFは、Fが左に1ビット位置ずつシフトされ、右端のビット位置に「ホット1」が付加されて、2つの3-2加算器CSA2とCSA3に供給される。CSA2にはEとFの他に $r h (D_2)$ も供給される。したがって、CSA2の結果は和GとキャリーHであり、これらは加算されると、Hが左に1ビット位置ずつシフトされ、右端のビット位置に「ホット1」が供給されて、 $B_2 + D_2 - (B_1 + D_1)$ を生成することになる。その間にEとFの左端の40ビット位置が、CSA3に送られて、40個の1と加算される。これによって、折返し条件及び再マップ条件の場合に $B_2 + D_2 - (B_1 + D_1)$ の計算を補正する際に使用される値が生成される(図5及び6及び本発明の挙動を指定する以下の諸式のεとδ)。CSA2からのGとHは、2-1加算器によって加算されるのではなく、CSA7に供給され、そこでこれらの値にLが加えられて、和oとキャリーωを生成する。それに並列に、G(56:63)とH(57:63) || 0 がキャリー生成器に供給されて、標準のキャリー先読み技法によってκ56を生成する。その間にCSA4がCSA2の和の選択された組合せをCSA3からのキャリーと加算して、和vとキャリーNを生成し、CSA5がCSA3の和をCSA2のキャリーと加算して、和vとキャリーΓを生成し、CSA6がCSA3の和をそのキャリーと加算して、和τとキャリーTを生成する。これらの値は、折返し条件及び再マップ条件での補正に必要なものである。

10

【 数142 】

60

*マップ条件での補正に必要なものである。

【 0 2 1 5 】様々な補正のケース用の数式142を生成する準備として、CSA4、CSA5、CSA6、CSA7によってそれぞれ生成された和v、v、τ、oとキャリーN、Γ、T、ωの様々な組み合わせが、κ56と共に排他的ORされる。これと並行して、o(56:63)とω(57:63) || 0 がキャリー生成器に供給されてφ56を生成し、これを数式143に従ってκ56及びω56と組み合わせることによってC56が生成できる。

【 数143 】

 $S \pm o$

【 数143 】

$$C_{56} = (\omega_{56} \vee x_{56}) \vee \phi_{56}$$

20
20

【 0 2 1 6 】この計算は、図5及び6を見るとわかるように、一連の2元排他的ORによって実行される。次いですべての排他的ORの適当な出力が57W-ANDに供給されて、表6に要約した6つのケースについてのMVC型オーバーラップO MVCを生成する。この適当な値は、図5及び6にOVERLAPとして示す正しいMVC型オーバーラップ条件を生成するため、図4に示す復号器によって生成されるCOR(1:5)によって選択される。次いでOVERLAPが、後続サイクルで実行すべく2つのマイクロ命令のうちの1つを選ぶために制御記憶機構の次のアドレス指定論理機構に提示するため、分岐低マルチプレクサ(図5及び6には示さず)に供給される。OVERLAPはまた、その命令サイクル中の後の時間にこの条件が後続のマイクロ命令にとって利用できるようにするためにラッチされる。このラッチは、EIRレジスタから取り出されるNEWIDビットによってリセットされる。このNEWID_EIRは制御記憶論理機構によってセットアップされる。このNEWID信号のタイミングは、次の通りである。

ID|AG|EX

mD|AG|EX

...mD|AG|EX (AG中の早期にGate_Cnd_EOPを生成する)

mD|AG|EX (CEOP -> AIR中で有効なNEWID)

ID|AG|EX (EIRは活動状態のNEWIDを有し、それを使用する)

【 0 2 1 7 】例外のためにマイクロコード式ルーチンの最中にマイクロコード式例外ハンドラが呼び出された場合、NEWIDは、マイクロコード式例外ハンドラの終りに有効にならない。NOT_NEWD_EIRまたはPIPELINE_HOLDが活動状態の場合はOVERLAPラッチが保持され、そうでない場合は新しい値がラッチされる。すべてのオーバーラップBRLO条件のラッチングは制御記憶機構内で実行される。詳細についてはそのワークブックを参照のこと。オーバーラップ条件の非ラッチ版は論理図に示してある。

50

ID|AG|EX (AG中の早期にGate_Cnd_EOPを生成する)

mD|AG|EX (CEOP -> AIR中で有効なNEWID)

ID|AG|EX (EIRは活動状態のNEWIDを有し、それを使用する)

1 h : B2を格納するLHレジスタのi番のビット。

1 a : B1を格納するLAレジスタのi番のビット。

r h : D2を格納するRHレジスタのi番のビット。これは12ビット・レジスタであると仮定し、そのように番号をつけてある。このレジスタを加数として使用する前に左に0を56個連結すると、AGENが発生する。

r a : D1を格納するRAレジスタのi番目のビット。

61

これは12ビット・レジスタであると仮定し、そのように番号をつけてある。このレジスタを加数として使用する前に左に0を56個連結すると、AGE Nが発生す *

$$e_i = lh_i \vee la_i$$

$$e_i = lh_i \vee \overline{la_i} \vee \overline{ra_{(i-52)}}$$

$$f_i = lh_i + \overline{la_i} = \overline{lh_i} la_i$$

$$f_i = lh_i \overline{la_i} + lh_i \overline{ra_{(i-52)}} + \overline{la_i} \overline{ra_{(i-52)}}$$

62

【 数144】

$$0 \leq i \leq 51$$

$$52 \leq i \leq 63$$

$$1 \leq i \leq 51$$

$$52 \leq i \leq 63$$

【 数145】

10

$$g_i = e_i \vee f_{i+1}$$

$$0 \leq i \leq 51$$

$$g_i = e_i \vee f_{i+1} \vee rh_{(i-52)}$$

$$52 \leq i \leq 62$$

$$g_{63} = \overline{e_{63}} \vee rh_{(11)}$$

$$h_i = e_i f_{i+1}$$

$$1 \leq i \leq 51$$

$$h_i = e_i f_{i+1} + e_i rh_{(i-52)} + f_{i+1} rh_{(i-52)}$$

$$52 \leq i \leq 62$$

$$h_{63} = e_{63} + rh_{(11)}$$

【 数146】

$$\delta_i = \overline{g_i}$$

$$0 \leq i \leq 39$$

$$e - n_i = \overline{e_i + f_{i+1}}$$

$$1 \leq i \leq 39$$

【 数147】

$$\sigma_i = g_i \vee h_{i+1}$$

$$0 \leq i \leq 55$$

$$\sigma_i = g_i \vee h_{i+1} \vee l_{(i-56)}$$

$$56 \leq i \leq 62$$

$$\sigma_{63} = g_{63} \vee \overline{l_7}$$

$$\omega_i = g_i h_{i+1}$$

$$1 \leq i \leq 55$$

$$\omega_i = g_i h_{i+1} + g_i l_{(i-56)} + h_{i+1} l_{(i-56)}$$

$$56 \leq i \leq 62$$

$$\omega_{63} = g_{63} + l_7$$

【 数148】

$$\tau_i = g_i ; \forall e - n_{i+1}$$

$$0 \leq i \leq 38$$

$$T_i = \overline{g_i + e - n_{i+1}}$$

$$1 \leq i \leq 38$$

【 数149】

40

$$\gamma_i = \overline{g_i} \vee \overline{h_{i+1}}$$

$$i = 32, 39$$

$$\Gamma_i = \overline{g_i + h_{i+1}}$$

$$i = 32, 39$$

【 数150】

$$v_i = \overline{g_i} \vee e - n_{i+1}$$

$$i = 31, 38$$

$$N_i = \overline{\overline{g_i} + e - n_{i+1}}$$

$$i = 31, 38$$

【 数151 】

$$\begin{aligned}
 (\kappa H)_i &= g_i \forall h_{i+1} & 56 \leq i \leq 63 \\
 (\kappa G)_{i_N} &= \overline{g_i \ h_{i+1}} & 56 \leq i \leq 62 \\
 (\kappa T)_{i_N} &= \overline{g_i + h_{i+1}} & 56 \leq i \leq 62 \\
 (\kappa C)_{63_N} &= \overline{g_{63}} \\
 \\
 (\kappa G)_{i+1}^{i+1} &= \overline{(\kappa G)_{i_N} ((\kappa T)_{i_N} + (\kappa G)_{i+1_N})} & 56 \leq i \leq 61 \\
 (\kappa T)_{i+1}^{i+1} &= \overline{(\kappa T)_{i_N} + (\kappa T)_{i+1_N}} & 56 \leq i \leq 61 \\
 (\kappa C)_{62_N} &= \overline{(\kappa G)_{62_N} ((\kappa T)_{62_N} + (\kappa C)_{63_N})} \\
 (\kappa C)_{63} &= \overline{(\kappa C)_{62_N}} \\
 (\kappa G)_{i+2_N} &= \overline{(\kappa G)_{i+1}^{i+1} + (\kappa T)_{i+1}^{i+1} (\kappa G)_{i+2}^{i+2}} & 56 \leq i \leq 59 \\
 (\kappa T)_{i+2}^{i+2} &= \overline{(\kappa T)_{i+1}^{i+1} (\kappa T)_{i+2}^{i+2}} & 56 \leq i \leq 59 \\
 (\kappa C)_{i_N} &= \overline{(\kappa G)_{i+1}^{i+1} + (\kappa T)_{i+1}^{i+1} (\kappa C)_{i+2}} & i = 60, 61 \\
 (\kappa C)_{i_N} &= \overline{(\kappa C)} & i = 62, 63 \\
 \kappa_i &= \overline{(\kappa G)_{i+3}^{i+3} ((\kappa T)_{i+3}^{i+3} + (\kappa C)_{i+4_N})} & 57 \leq i \leq 59 \\
 (\kappa C)_i &= \overline{(\kappa C)_{i_N}} & 60 \leq i \leq 63 \\
 \kappa_{56} &= \overline{(\kappa G)_{56_N} ((\kappa T)_{56_N} + (\kappa C)_{60_N})} \\
 S_i &= (\kappa C)_{i+1} \forall (\kappa H)_i & 57 \leq i \leq 63 \\
 S_{56} &= \kappa_{57} \forall (\kappa H)_{56} \\
 S_{63_N} &= \overline{S_{63}}
 \end{aligned}$$

【 数152 】

$$\begin{aligned}
 (\phi G)_{i_N} &= \overline{\sigma_i \omega_{i+1}} & 56 \leq i \leq 62 \\
 (\phi T)_{i_N} &= \overline{\sigma_i + \omega_{i+1}} & 56 \leq i \leq 62 \\
 (\phi C)_{63_N} &= \overline{\sigma_{63} \omega_{64}} \\
 (\phi G)_{i+1}^{i+1} &= \overline{(\phi G)_{i_N} ((\phi T)_{i_N} + (\phi G)_{i+1_N})} & i = 56, 58, 60 \\
 (\phi T)_{i+1}^{i+1} &= \overline{(\phi T)_{i_N} + (\phi T)_{i+1_N}} \\
 (\phi C)_{62_N} &= \overline{(\phi G)_{62_N} ((\phi T)_{62_N} + (\phi C)_{63_N})} \\
 (\phi G)_{56_N} &= \overline{(\phi G)_{56_N} + (\phi T)_{56_N} (\phi G)_{56_N}} \\
 (\phi T)_{56_N} &= \overline{(\phi T)_{56_N} (\phi T)_{56_N}} \\
 (\phi C)_{60_N} &= \overline{(\phi G)_{60_N} + (\phi T)_{60_N} (\phi C)_{60_N}} \\
 \phi_{56} &= \overline{(\phi G)_{56_N} ((\phi T)_{56_N} + (\phi C)_{60_N})}
 \end{aligned}$$

【 数153 】

XR 及び XR8

$C_{56} = \kappa_{56} \forall \omega_{56} \forall \phi_{56}$

* 【 数155 】

$S_{\text{OCOR1}}^t = (\sigma_0 \forall \omega_1) (\sigma_1 \forall \omega_2) \dots (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$

$O_{\text{MVCCOR1}} = S_{\text{OCOR1}}^t C_{56}$

【 数154 】

$S_{\text{OCOR2}}^t = (\sigma_0 \forall \omega_1) (\sigma_1 \forall \omega_2) \dots (\sigma_{35} \forall \omega_{35}) (\sigma_{36} \forall \omega_{37}) (\sigma_{37} \forall N_{38}) (v_{38} \forall T_{39}) (y_{39} \forall \omega_{40}) (\sigma_{40} \forall \omega_{41}) (\sigma_{41} \forall \omega_{42}) \dots (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$ $(\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$

$O_{\text{MVCCOR2}} = S_{\text{OCOR2}}^t C_{56}$

【 数156 】

$S_{\text{OCOR3}}^t = (\tau_0 \forall T_1) (\tau_1 \forall T_2) \dots (\tau_{35} \forall T_{37}) (\tau_{37} \forall T_{38}) (T_{38} \forall \Gamma_{39}) (y_{39} \forall \omega_{40}) (\sigma_{40} \forall \omega_{41}) (\sigma_{41} \forall \omega_{42}) \dots (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$

$O_{\text{MVCCOR3}} = S_{\text{OCOR3}}^t C_{56}$

【 数157 】

$$S_{\text{OCOR4}}^t = (\sigma_0 \vee \omega_1) (\sigma_1 \vee \omega_2) \dots (\sigma_{28} \vee \omega_{29}) (\sigma_{29} \vee \omega_{30}) (\sigma_{30} \vee N_{31}) (\nu_{31} \vee \Gamma_{32}) (\gamma_{32} \vee \omega_{33}) (\sigma_{33} \vee \omega_{34}) (\sigma_{34} \vee \omega_{35}) \dots \\ (\sigma_{34} \vee \omega_{35}) (\sigma_{35} \vee \kappa_{36})$$

$$O_{MVCCOR4} = S_{\text{OCOR4}}^t C_{36}$$

【 数158 】

$$S_{\text{OCOR5}}^t = (\tau_0 \vee T_1) (\tau_1 \vee T_2) \dots (\tau_{28} \vee T_{29}) (\tau_{29} \vee T_{30}) (\tau_{30} \vee T_{31}) (\tau_{31} \vee \Gamma_{32}) (\gamma_{32} \vee \omega_{33}) (\sigma_{33} \vee \omega_{34}) (\sigma_{34} \vee \omega_{35}) \dots (\sigma_{34} \vee \omega_{35}) (\sigma_{35} \vee \kappa_{36})$$

$$O_{MVCCOR5} = S_{\text{OCOR5}}^t C_{36}$$

【 数159 】

S A R 1 H I 12ビット加算検出

$SAR1H_i = \overline{Ia_i \vee ra_{i-53}}$	$52 \leq i \leq 55$
$SAR1G_i_N = \overline{Ia_i \vee ra_{i-52}}$	$52 \leq i \leq 62$
$SAR1T_i_N = \overline{Ia_i + ra_{i-52}}$	$52 \leq i \leq 62$
$SAR1C_{53}_N1 = \overline{Ia_{53} \vee ra_{11}}$	
$SAR1H_i_N = \overline{SAR1H_i}$	$52 \leq i \leq 55$
$(SAR1G)_i^{+1} = \overline{(SAR1G)_i_N ((SAR1T)_i_N + (SAR1G)_{i+1}_N)}$	$52 \leq i \leq 61$
$(SAR1T)_i^{+1} = \overline{((SAR1T)_i_N + (SAR1T)_{i+1}_N)}$	$52 \leq i \leq 61$
$(SAR1C1)_{52} = \overline{(SAR1G)_{52}_N ((SAR1T)_{52}_N + (SAR1C)_{53}_N1)}$	
$(SAR1C1)_{53} = \overline{(SAR1C)_{53}_N1}$	
$(SAR1G)_i^{+3}_N = \overline{(SAR1G)_i^{+1} + (SAR1T)_i^{+1} (SAR1G)_{i+2}^{+1}}$	$52 \leq i \leq 59$
$(SAR1T)_i^{+3}_N = \overline{(SAR1T)_i^{+1} (SAR1T)_{i+2}^{+1}}$	$52 \leq i \leq 59$
$(SAR1C)_i_N2 = \overline{(SAR1G)_i^{+1} + (SAR1T)_i^{+1} (SAR1C1)_{i+2}}$	$i = 60, 61$
$(SAR1C)_i_N2 = \overline{(SAR1C1)_i}$	$i = 62, 63$
$(SAR1G)_i^{+7} = \overline{(SAR1G)_i^{+3}_N ((SAR1T)_i^{+3}_N + (SAR1G)_{i+4}^{+1}_N)}$	$52 \leq i \leq 55$
$(SAR1T)_i^{+7} = \overline{(SAR1T)_i^{+3}_N + (SAR1T)_{i+4}^{+1}_N}$	$52 \leq i \leq 55$
$(SAR1C)_{56} = \overline{(SAR1G)_{56}_N ((SAR1T)_{56}_N + (SAR1C)_{60}_N2)}$	56
$(SAR1C)_i = \overline{(SAR1C)_i_N2}$	$60 \leq i \leq 63$
$(SAR1C)_i_N = \overline{(SAR1G)_i^{-7} + (SAR1T)_i^{+7} (SAR1C)_{i+8}}$	$52 \leq i \leq 55$
$(SAR1C)_{56}_N = \overline{(SAR1C)_{56}}$	
$SAR1_i_N = (SAR1C)_{i+1}_N \vee SAR1H_i$	$52 \leq i \leq 55$
$SAR1_i = (SAR1C)_{i+1}_N \vee SAR1H_i_N$	$52 \leq i \leq 55$
$SAR1_52_to_55_ZEROS = (SAR1_{52}_N) (SAR1_{53}_N) (SAR1_{54}_N) (SAR1_{55}_N)$	
$SAR1_52_to_55_ONES = (SAR1_{52}) (SAR1_{53}) (SAR1_{54}) (SAR1_{55})$	
$(SAR1C)_{52} = \overline{(SAR1C)_{52}_N}$	

【 数160 】

SAR1HI、SAR1LO、WRAP1 檢出論理

$$la_i_N = \overline{la_i} \quad 33 \leq i \leq 51$$

$$la_{40_TO_51_ZEROS} = la_{40_N} la_{41_N} \dots la_{50_N} la_{51_N}$$

$$la_{40_TO_50_ONES} = la_{40} la_{41} \dots la_{50}$$

$$la_{40_TO_51_ONES} = la_{40_TO_50_ONES} la_{51}$$

$$la_{40_TO_50_ONES_A_51_ZERO} = la_{40_TO_50_ONES} la_{51_N}$$

$$la_{33_TO_39_ZEROS} = la_{33_N} la_{34_N} \dots la_{38_N} la_{39_N}$$

$$la_{33_TO_39_ONES} = la_{33} la_{34} \dots la_{38} la_{39}$$

$$la_{33_TO_50_ONES_A_51_ZERO} = la_{33_TO_39_ONES} la_{40_TO_50_ONES_A_51_ZERO}$$

$$la_{33_TO_51_ZEROS} = la_{33_TO_39_ZEROS} la_{40_TO_51_ZEROS}$$

$$la_{33_TO_51_ONES} = la_{33_TO_39_ONES} la_{40_TO_51_ONES}$$

$$SAR1_TO_51_0S = 24_BIT_MD la_{40_TO_51_ZEROS} + 31_BIT_MD la_{33_TO_51_ZEROS}$$

$$SAR1_TO_51_1S = 24_BIT_MD la_{40_TO_51_ONES} + 31_BIT_MD la_{33_TO_51_ONES}$$

$$SAR1_TO_50_1S_A_51_0 = 24_BIT_MD la_{40_TO_50_ONES_A_51_ZERO} +$$

$$31_BIT_MD la_{33_TO_50_ONES_A_51_ZERO}$$

$$SAR1LO = [(SAR1C)_{s2_N} SAR1_TO_51_0S + (SAR1C)_{s2} SAR1_TO_51_1S] SAR1_52_TO_55_ZEROS$$

$$SAR1HI = [(SAR1C)_{s2_N} SAR1_TO_51_1S + (SAR1C)_{s2} SAR1_TO_50_1S_A_51_0] SAR1_52_to_55_ONES$$

$$WRAP1 = (SAR1C)_{s2} SAR1_TO_51_1S$$

$$WRAP1_N = \overline{WRAP1}$$

【 数1 6 1 】

S A R 2 H I 12ビット加算検出

$SAR2H_i = \overline{lh_i \cdot rh_{(i-52)}}$	$52 \leq i \leq 55$
$SAR2G_{i-N} = \overline{lh_i \cdot rh_{(i-52)}}$	$52 \leq i \leq 62$
$SAR2T_{i-N} = \overline{lh_i + rh_{(i-52)}}$	$52 \leq i \leq 62$
$SAR2C_{63-N1} = \overline{lh_{63} \cdot rh_{11}}$	
$SAR2H_{i-N} = \overline{SAR2H_i}$	$52 \leq i \leq 55$
$(SAR2G)_i^{+1} = \overline{(SAR2G)_i-N ((SAR2T)_i-N + (SAR2G)_{i+1-N})}$	$52 \leq i \leq 61$
$(SAR2T)_i^{+1} = \overline{((SAR2T)_i-N + (SAR2T)_{i+1-N})}$	$52 \leq i \leq 61$
$(SAR2C1)_{62} = \overline{(SAR2G)_{62-N} ((SAR2T)_{62-N} + (SAR2C)_{63-N1})}$	
$(SAR2C1)_{63} = \overline{(SAR2C)_{63-N1}}$	
$(SAR2G)_i^{+2-N} = \overline{(SAR2G)_i^{+1} + (SAR2T)_i^{+1} (SAR2G)_{i+2}}$	$52 \leq i \leq 59$
$(SAR2T)_i^{+2-N} = \overline{(SAR2T)_i^{+1} (SAR2T)_{i+2}}$	$52 \leq i \leq 59$
$(SAR2C)_i-N2 = \overline{(SAR2G)_i^{+1} + (SAR2T)_i^{+1} (SAR2C1)_{i+2}}$	$i = 60, 61$
$(SAR2C)_i-N2 = \overline{(SAR2C1)}$	$i = 62, 63$
$(SAR2G)_i^{+3} = \overline{(SAR2G)_i^{+2-N} ((SAR2T)_i^{+1} N + (SAR2G)_{i+2-N})}$	$52 \leq i \leq 55$
$(SAR2T)_i^{+3} = \overline{(SAR2T)_i^{+2-N} + (SAR2T)_{i+2-N}}$	$52 \leq i \leq 55$
$(SAR2C)_{56} = \overline{(SAR2G)_{56-N} ((SAR2T)_{56-N} + (SAR2C)_{60-N2})}$	56
$(SAR2C)_i = \overline{(SAR2C)_i-N2}$	$60 \leq i \leq 63$
$(SAR2C)_i-N = \overline{(SAR2G)_i^{+2} + (SAR2T)_i^{+2} (SAR2C)_{i+8}}$	$52 \leq i \leq 55$
$(SAR2C)_{56-N} = \overline{(SAR2C)_{56}}$	
$SAR2_{i-N} = \overline{(SAR2C)_{i+1-N} \vee SAR2H_i}$	$52 \leq i \leq 55$
$SAR2_i = \overline{(SAR2C)_{i-1-N} \vee SAR2H_i}$	$52 \leq i \leq 55$
$SAR2_52_to_55_ZEROs = \overline{(SAR2_{52-N}) (SAR2_{53-N}) (SAR2_{54-N}) (SAR2_{55-N})}$	
$SAR2_52_to_55_ONES = \overline{(SAR2_{52}) (SAR2_{53}) (SAR2_{54}) (SAR2_{55})}$	
$(SAR2C)_{52} = \overline{(SAR2C)_{52-N}}$	

【数162】

63

SAR2HI、SAR2LO、WRAP2 検出論理

64

$lh_i_N = \overline{lh_i}$ $33 \leq i \leq 51$

$lh_{40_TO_51_ZEROS} = lh_{40_N} lh_{41_N} \dots lh_{50_N} lh_{51_N}$

$lh_{40_TO_50_ONES} = lh_{40} lh_{41} \dots lh_{50}$

$lh_{40_TO_51_ONES} = lh_{40_TO_50_ONES} lh_{51}$

$lh_{40_TO_50_ONES_A_51_ZERO} = lh_{40_TO_50_ONES} lh_{51_N}$

$lh_{33_TO_39_ZEROS} = lh_{33_N} lh_{34_N} \dots lh_{39_N} lh_{40_N}$

$lh_{33_TO_39_ONES} = lh_{33} lh_{34} \dots lh_{39} lh_{40}$

$lh_{33_TO_50_ONES_A_51_ZERO} = lh_{33_TO_39_ONES} lh_{40_TO_50_ONES_A_51_ZERO}$

$lh_{33_TO_51_ZEROS} = lh_{33_TO_39_ZEROS} lh_{40_TO_51_ZEROS}$

$lh_{33_TO_51_ONES} = lh_{33_TO_39_ONES} lh_{40_TO_51_ONES}$

$SAR2_TO_51_OS = 24_BIT_MD lh_{40_TO_51_ZEROS} + 31_BIT_MD lh_{33_TO_51_ZEROS}$

$SAR2_TO_51_IS = 24_BIT_MD lh_{40_TO_51_ONES} + 31_BIT_MD lh_{33_TO_51_ONES}$

$SAR2_TO_50_IS_A_51_0 = 24_BIT_MD lh_{40_TO_50_ONES_A_51_ZERO} +$
 $31_BIT_MD lh_{33_TO_50_ONES_A_51_ZERO}$

$SAR2LO = [(SAR2C)_{s2} N SAR2_TO_51_OS + (SAR2C)_{s2} SAR2_TO_51_IS] SAR2_52_TO_55_ZEROS$

$SAR2HI = [(SAR2C)_{s2} N SAR2_TO_51_IS + (SAR2C)_{s2} SAR2_TO_50_IS_A_51_0] SAR2_52_TO_55_ONES$

$WRAP2 = (SAR2C)_{s2} SAR2_TO_51_IS$

$WRAP2_N = \overline{WRAP2}$

【数163】

C O R (1 : 5) 生成論理

$REMAP1_N = \overline{SAR2HI} \overline{SAR1LO}$

$REMAP2_N = \overline{SAR2LO} \overline{SAR1HI}$

$REMAP1 = \overline{REMAP1_N}$

$REMAP2 = \overline{REMAP2_N}$

$COR(1) = 64_BIT_MD + REMAP1_N WRAP1_N WRAP2_N + WRAP2 WRAP1$

$COR(2) = 24_BIT_MD REMAP1_N WRAP2_N WRAP1$

$COR(3) = 24_BIT_MD WRAP2 WRAP1_N + 24_BIT_MD REMAP1 WRAP1_N$

$COR(4) = 31_BIT_MD REMAP1_N WRAP2_N WRAP1$

$COR(5) = 31_BIT_MD WRAP2 WRAP1_N + 31_BIT_MD REMAP1 WRAP1_N$

【数164】

O V E R L A P 生成論理

$OVERLAP = COR(1) O_{HYCCOR1} + COR(2) O_{HYCCOR2} + COR(3) O_{HYCCOR3} + COR(4) O_{HYCCOR4} +$
 $COR(5) O_{HYCCOR5}$

【図面の簡単な説明】

【図1】記憶オペランドが破壊的記憶オペランド・オーバーラップを有する可能性のある、すべての独自ケースを示す図である。

【図2】記憶オペランドが破壊的記憶オペランド・オーバーラップを有しない、すべての独自ケースを示す図である。

バーラップを有しない、すべての独自ケースを示す図である。

【図3】記憶オペランドが破壊的記憶オペランド・オーバーラップを有しない、すべての独自ケースを示す図である。

65

【図4】破壊的記憶オペランド・オーバーラップを予測するのに必要なハードウェアの概要を示す図である。

【図5】図3の概要に含まれるオーバーラップ予測論理ブロックの詳細を示す図である。

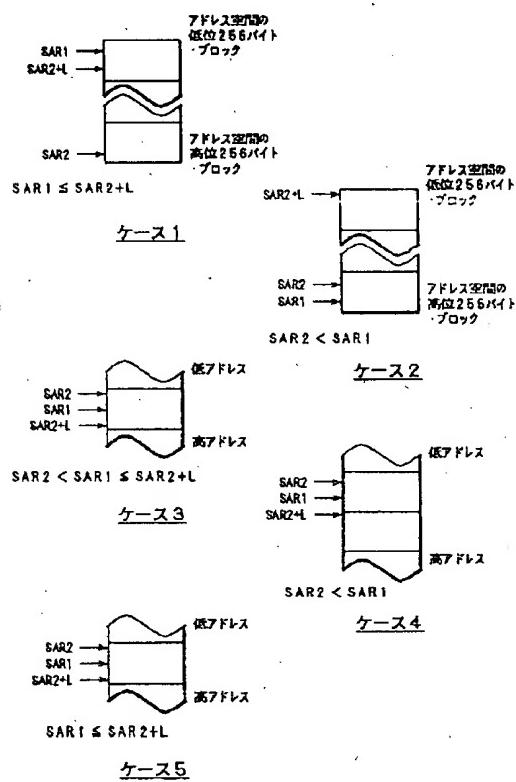
【図6】図3の概要に含まれるオーバーラップ予測論理ブロックの詳細を示す図である。

【図7】宛先記憶オペランド・アドレスが仮想アドレス空間の最高位256バイト・ブロックに常駐し、仮想アドレス空間の最低位256バイト・ブロックに常駐する

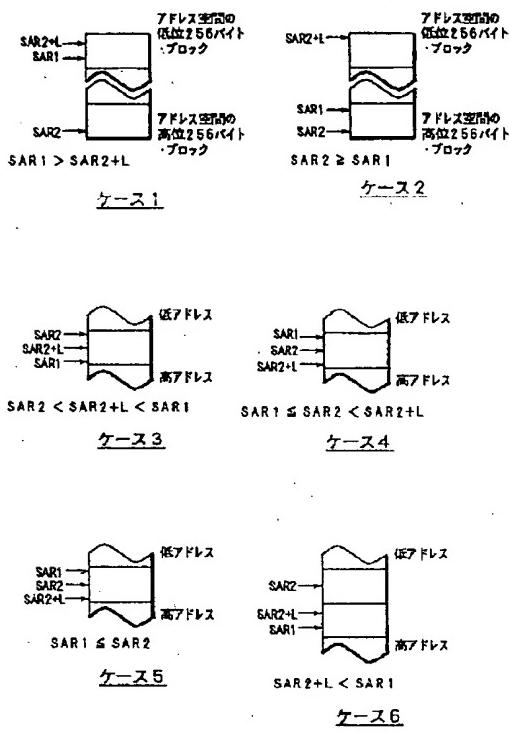
かどうか、および宛先記憶オペランド・アドレスの計算中に折返しが発生する条件を決定するための論理の詳細を示す図である。

【図8】ソース記憶オペランド・アドレスが仮想アドレス空間の最高位256バイト・ブロックに常駐し、仮想アドレス空間の最低位256バイト・ブロックに常駐するかどうか、およびソース記憶オペランド・アドレスの計算中に折返しが発生する条件を決定するための論理の詳細を示す図である。

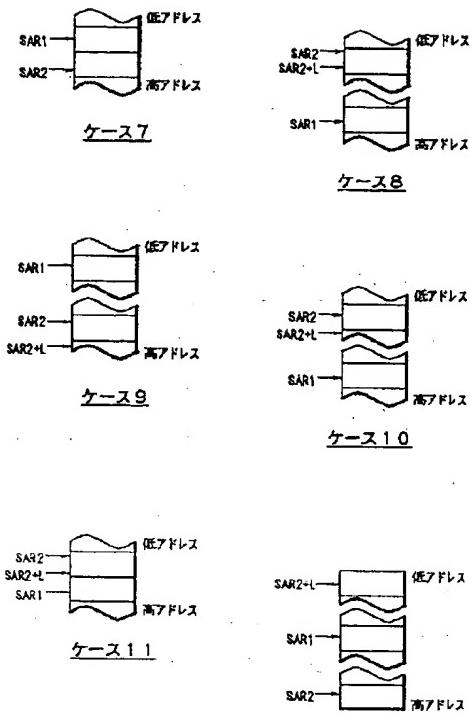
【図1】



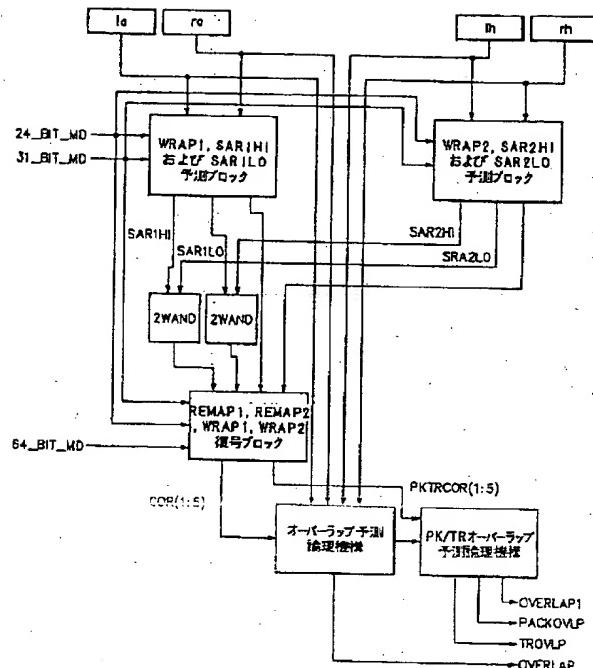
【図2】



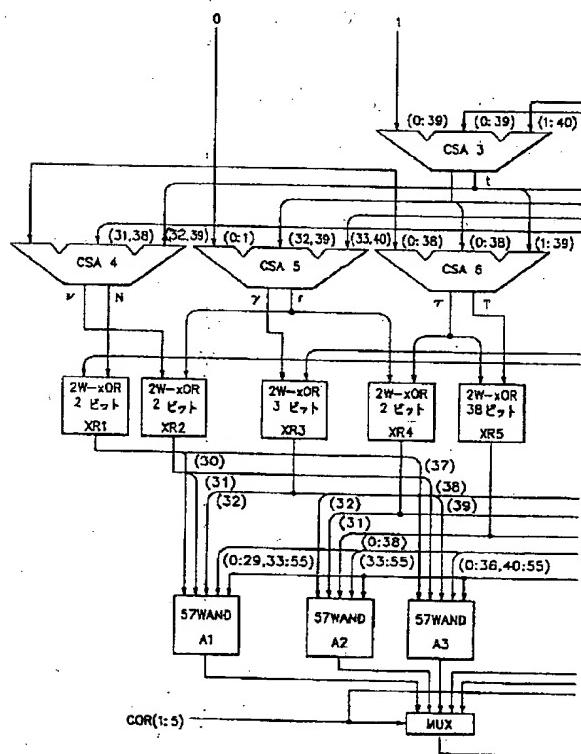
【 図3 】



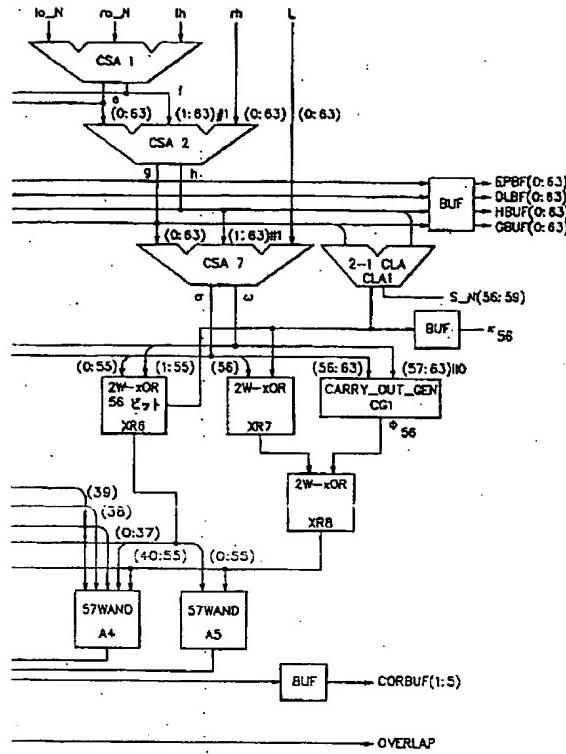
【 図4 】



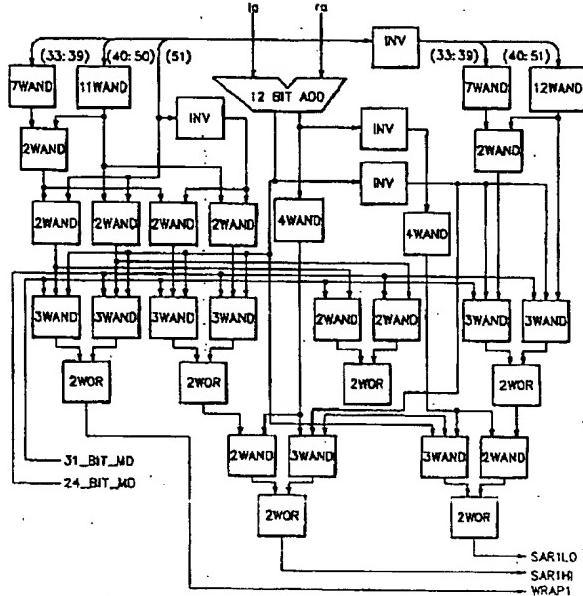
【 図5 】



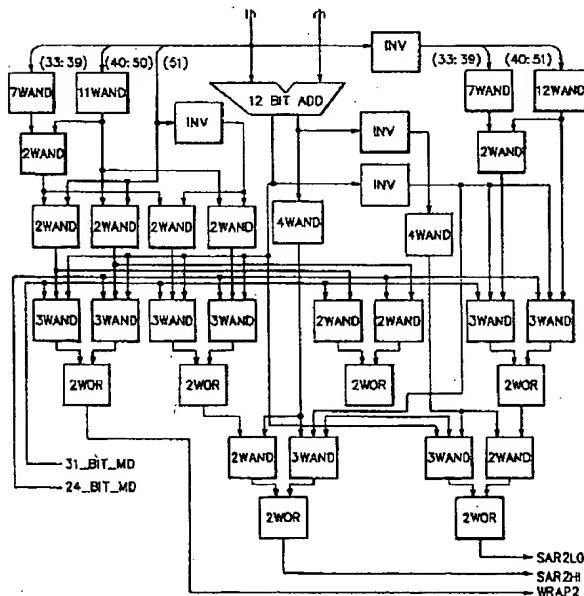
【 図6 】



【 図7 】



【 図8 】



フロントページの続き

(72)発明者 スタマティス・ヴァッシリアディス
アメリカ合衆国13850、ニューヨーク州ヴ
エスター、ヴェスター・ロード 717